

the world of

68' micros

Support for Motorola Processors

December 1994

Vol. 2 Number 4

\$4.50 Canada, \$4.00 US

NEWS FLASH...

UpTime and 68' micros merge!

01 NOV 94 - In an agreement to promote good will and solidarity in the CoCo /OS-9 community, FARNA Systems and JWT enterprises reached an agreement to merge the two publications and continue written support for subscribers. JWT decided to stop publication with the September issue. FARNA will pick up the remaining subscriptions at the rate of one issue of "68' micros" for every two issues of "UpTime" remaining. Those with subscriptions to both publications will have their "68' micros" subscriptions extended by one issue for every two issues of "UpTime" remaining. Subscriptions will resume as soon as JWT finishes compiling customer accounts and forwards them to FARNA.

In this issue:

Colin McKays' Atlanta Fest Report!
NEW! Disk Basic Support Column
Don't forget to read the editorial in
this issue for more news!

CONTENTS

<i>The Editor Speaks</i>	2
<i>Letters to the Editor</i>	3
<i>MM/I Update</i>	3
(column) <i>David Graham</i>	
<i>5th Ann. Atlanta CoCo Fest</i>	4
(article) <i>Colin McKay</i>	
<i>BASIC in Color</i>	7
(column) <i>Fred Remin</i>	
<i>Industrial OS-9 User...</i>	9
(G-Windows series) <i>Ed Gresick</i>	
<i>Trek</i>	10
(program) <i>Pete Blackwell</i>	
<i>The Hardware Hacker</i>	11
(column) <i>Dr. Marty Goodman</i>	
<i>Operating System-Nine</i>	13
(column) <i>Russell Hoffman</i>	
<i>OS-9/OSK Answers!</i>	16
(column) <i>Joel Hegberg</i>	
<i>WILDCARD.BAS</i>	19
(program) <i>C.R. Connolly</i>	
<i>Programming in "C"</i>	20
(column) <i>P.J. Ponzo</i>	
<i>Basic99 In Easy Steps</i>	22
(series) <i>Chris Dekker</i>	
<i>micronotes</i>	25
<i>Advertiser's Index</i>	29

NOTICE: AS OF 01 JANUARY 1995,
SUBSCRIPTION RATES WILL BE:
US: \$25/year, \$13/6 mths
Overseas: \$45/year, \$23/6 mths (air)
\$37/year, \$18/6 mths (surface)
APO/FPO: \$30/year, \$16/6 mths
(1st class US mail)
Canada/Mexico: NO CHANGE

POSTMASTER:

If undeliverable return to:
FARNA Systems PB
Box 321
Warner Robins, GA 31099

Address Correction Requested

the world of 68' micros

Published by:
FARNA Systems
P.O. Box 321
Warner Robins, GA 31099-0321

Editor: F. G. Swygart

Subscriptions:

\$25/year (8 issues) US; \$30/year for Canada/Mexico (\$13US, \$16C/M for six months- four issues). Overseas \$45/year (\$23 for four issues) AIR; \$37/year, \$18 six months for surface mail.
microdisk: \$40 per year, \$21 six months, or \$6 per issue. Overseas add \$10/year, \$5/six months, \$1/single issue for air mail delivery. microdisk contains programs and source listings from each magazine; not stand-alone.

Advertising Rates:

\$15 1/6 page, \$20 1/4 page, \$35 1/2 page, \$60 full page, copy ready. Add \$10 for special placement, \$10 for typesetting (\$5 1/4 or less). Dot matrix will be typeset if deemed unacceptable and submitter billed. 10% discount for four or more appearances.

All trademarks/names property of their respective owners.

The publisher welcomes any and all contributions. Submission constitutes warranty on part of the author that the work is original and not copywritten by another party. All opinions expressed herein are those of the individual writers, not necessarily the publisher or editor. FARNA Systems reserves the right to edit or reject any submitted material without explanation. Renumeration discussed on an individual basis.

Back issues are \$4 per copy. Overseas add \$1 each for airmail delivery.

Newsstand/bulk orders available. Dealers should contact the publisher for details.

Problems with delivery, change of address, subscriptions, or advertisers should be sent to the publisher with a short description.

The publisher is available for comment via e-mail at microed@zog.com or dsrtfox@Delphi.com. The Delphi CoCo and OS-9 SIGs on Delphi are also frequented (The Delphi SIGs are still sponsored by Falsoft).

ENTIRE CONTENTS COPYRIGHT
1994, FARNA Systems

The editor speaks...

F.G. Swygart

I'm sure many of you read the "headlines" on the cover, they are very hard to miss! As I write this, I haven't received the subscriber list from UpTime, so I can't comment on how many new subscribers this will bring in.

During an e-mail conversation with Jordan Tsvetkov, UpTime editor/publisher, he mentioned that he was considering folding UpTime this fall. He had only brought it out as a supplement to the Rainbow, mainly for advertisers who couldn't afford Rainbow's high rates. With the Rainbow gone and the more reasonable rates of 68' micros and The International OS-9 Underground, Jordan felt that UpTime had served its purpose. I decided it might be good for 68' micros to absorb those subscribers rather than let them get "lost" in the wayside. So we forged an agreement where I would continue UpTime subscriptions at my cost, which handily worked out to one issue of 68' micros for every two issues of UpTime. On a page count basis, this works out very well for former UpTime subscribers, as UT averaged 8 pages per issue and 268'm averages 30 (and the page contents are more dense also).

I hope you saw the notice in the contents column. First, the price increase has nothing to do with the UpTime merger. It is simply due to the costs and time necessary to produce a magazine of reasonable quality. I do hope you understand (I work for about \$2.50 an hour to bring you this magazine!). I love the CoCo and OS-9 in all its forms, but I can't put the magazine out at a loss. On the other hand, I don't expect to make much either. Nicely breaking even, with just a little left over, will suit me fine! There should be no more price increases for quite a while, unless the US Postal Service has a major price hike (over .32 for 1st class letters) or subscriptions drop below 200. In the later case, all will have to be mailed via first class postage, which means an additional .50 per issue (\$4 per year). That means rates would have to increase to \$28 per year. If this happens, the increase will be effective with the next issue, which means all subscribers would receive renewal notices one issue earlier than the year before. I'll take absorb the remaining \$1 per subscription.

Now I have to get something off my chest. I make every effort to keep readers informed about what is going with 68' micros, including some financial information on occasion. This is YOUR magazine!!! But I NEED feedback and even criticism from you, so WRITE!

I was really disturbed by two notes recently received in the mail. Both were from people who let their subscriptions expire. I had sent

renewal notices asking for a reason if subscriptions were not renewed. These two decided to let me know.

What really upset me was the fact that they didn't write earlier, when something could be done about their complaints. It takes at least two issues for changes to be incorporated in most cases. Some I can make effective immediately, depending on how much work has progressed on the current issue and how close it is to going out.

These people did no one any good, especially not themselves, by dropping their subscriptions. If they had written and let me know what they weren't getting, they could have easily been accommodated. For the record, the complaints were not enough Color Basic specific support and no more Swap-Shop.

The last issue saw reaffirmation of Color Basic support. I'll have program listings in every issue from now on while continuing the current amount of OS-9/OSK support.

I simply was not getting enough response to have a dedicated classifieds section. Classified ads will appear in "micro notes" (formerly "Micro News") from now on as readers send them in.

But those two who let their subscriptions expire rather than write and let me know they had complaints will never know of these changes. And subscription rates may rise in the future for all those who continue because those two (and others who won't write and let me know their needs) are no longer subscribers...

PROGRAMMING CONTEST!!!

FARNA Systems has set aside some cash and prizes (\$150 total!) for a programming contest!

ALL COMPUTER TYPES SUPPORTED BY 68'micros

ARE ELIGIBLE!

Send a disk with a running copy of the program as well as an ASCII listing of the source code or BASIC listing, running/installing instructions, and a description to FARNA Systems PC, Box 321, WR, GA 31099 by January 1st, 1995. Programs may be of any type.

**First place gets \$50 cash,
Second place gets \$25!**

Letters to the Editor

I have a question about ADOS: will it give you a logical 360/720K disk drive (40/80track double sided), or just two 180/360K (40/80 track single sided) disks back to back? Are there any other ROM tricks that will give you one logical 360K or 720K disk ala MS-DOS?

Now I would like to share some private thoughts on our little CoCo. I'm pleased that I own a fairly nice home, one that takes a good deal of my weekend time at least. Which do I think about more, home or CoCo? The question hardly needs to be asked! How can this be -- it's just a little machine, right? Just plastic, circuit boards and silica chips, right? Balance is a good idea -- we've got trees, a lawn, five dogs, and seven cats (anyone close by can have some!) to tend to. A 1982 concord and a Significant Other, thank God. Should balance be so hard?

Fortunately, I count my blessings in having met you, Zack Sessions, Tony DiStefano, Alan Huffman, Rick Ulland, and Colin McKay; the depth of your collective understanding and ability is awesome. Adding Bob van der Poel, Marty Goodman, and Matt Thompson, this is an incomparable collection of souls. Overall, I feel truly enriched to have met any of ya'll. You are the spirit and soul of the CoCo community.

Henry Harwell
2110 W. Roma Ave.
Phoenix, AZ 85015-4445

First of all, thanks for the kind words, Henry! I'm sure the other writers and vendors appreciate the comments also. I did edit your letter rather severely, as much of it was directed to me personally. I do hope you don't mind that I shared some of your comments with the readers though, especially those about the other vendors and writers.

As for ADOS: the problem is the CoCo disk system itself. There is only room for a maximum of 78 file entries, so even if you could address both sides of a disk at once (MS-DOS does it by alternating sides... a file is written to track 0 side 1 then track 0 side 2, track 1 side 1, track 1 side 2, etc.), you would be limited to 78 files total. Since most CoCo files are

small, that would sure be a waste of space! Note that this problem persists with 80 track drives. The only purpose I can personally see for using 80 track drives with ADOS is that they are already on the system for use with OS-9, which can take full advantage of the capacity. At least one standard 35/40 track drive would be needed even then.

Want to let you know I received Vol.2 #3 following our renewal "checkup". The bad news is that I never received Vol.2 #2. I hope you can rectify this omission.

I also have a question about RGB-DOS. I run a DS/DD 5.25" drive and two 3.5" 720K drives. I would like to partition the 720K drives into four 35 track drives each under DECB, much like RGB-DOS does with a hard drive. That would allow me much more efficient utilization of my disks with Max-10 and other DECB applications. As you know, Max-10 won't run under Hyper I/O or the double sided pokes.

Ray Watts
Box 574
Niantic, CT 06357

Ray, a copy of the missing issue is on its way! As far as I know, what you are asking is not possible. RGB-DOS only addresses the hard drive itself as multiple floppies, it wouldn't work with the floppy drives themselves. The only thing I can think of that would partially rectify your problem is Extended ADOS 3. It allows mixing of drive sizes, so you could have your 3.5" drives set up as two 360K drives. You can even have different configurations, so you could take the 5.25" off line and use the two 3.5" drives only (as the four allowable drive numbers).

< 268'm >

Letters are printed on a space available and popular subject matter basis. If you don't want your letter printed, or wish to withhold your address and/or name, please state so when writing. In some cases, letters are edited for space and/or clarity. If a personal reply is desired, please enclose an SASE.

MM/1 Update

David Graham

Greetings, and a Happy Holiday Season to you and yours! We at BlackHawk Enterprises, Inc. are continuing to make progress with new products for your MM/1. A beta version of SpareRibBS went into testing late in November and we hope to release in January, or perhaps as early as Christmas! SpareRibBS is the shareware version of John Donaldson's RibBS port, which will be followed in April by a more polished and capable commercial release of K-RibBS. John is going to be a busy camper during the next few months, as he will also be working on upgrading K-Term to add Compuserve-B protocol and improved ANSI screen controls, among other things yet to be determined. Look for us at the Chicago CoCoFest in April! We also have acquired sourcing for Sony's CPD-1302 monitor. No longer produced, these are refurbished units. At \$225 it is hard to find a better buy, as these monitors sport a .26 dot pitch and deliver an outstanding display on your MM/1! We can also offer tape backup drives for as low as \$175 used or \$425 new - though our sources for used tape drives are not as stable as that for the monitors.

Work on getting MM/1 I/O boards to production continues, with Kevin Pease, Ray Patterson and Joel Ewe working together to isolate problems with the MM/1 SCSI sub-system. Progress is being made, and we hope to have more news for you by next month. In the mean time -

Merry Christmas!!

< 268'm >



BlackHawk
Enterprises, Inc.
*Supporting the
MM/1...
now and in the future!*

The Fifth Annual Atlanta CoCoFest

Southern spotlight on the CoCo, OS-9, and OSK!

Colin McKay

ATLANTA, GA: Once again another Atlanta Fest has come and gone. For those who were unable to attend, I've prepared this report on the Fest. Those of you on electronic services should also keep your eyes open for Allen Huffman's Fest Report, for a slightly different perspective.

The Trip Down

This time around Ken Scales, Jim Pottage and I went down to the Atlanta Fest in Jim's car. Apart from an interesting conversation with US Customs ("Please pull into bay #2"), the trip down was uneventful (While the North American Free Trade Agreement, or NAFTA, has aided trade for big business, it has also imposed an additional burden in terms of paperwork for small businesses, such as Northern Xposure. Hardware in particular has become difficult to take across the border, though sending it by mail is relatively straight-forward). Twenty three hours later, we arrived at the hotel.

Attendance seemed down once again, but sales were still pretty good, especially for those who had new products available. Those without new products quickly discover that people will only buy a product once! No cheerleader convention at the hotel this year either (OK, so they were a little young, but it sounds good)! Business was steady on Saturday, and hectic Sunday morning, but Sunday afternoon was pretty quiet as people began the journey home.

Sincere thanks to the Atlanta Computer Society (ACS), our hosts for the Fest. The show, as always, was well run. Anyone thinking of putting on a show in their region should strongly consider contacting either ACS, or the Glenside Club, which hosts the Chicago Fests. Once again ACS negotiated a substantial discount on the regular hotel rates for those who stayed at the Holiday Inn.

The Vendors

Did my best to make it around to all the vendors this time, to see what they had to offer, and made purchases when I found interesting items. While my interests lie primarily with OS-9/68000 software, I tried to cover both DECB and OS-9/6809 products as well.

Twenty-six vendors were scheduled to attend, with a mixture of old and new products. Some quick changes were made when some vendors had to cancel at the last minute, and some additions were also made. This was the largest number of vendors ever at the Atlanta show. I've noted the vendors here,

along with product highlights from each one. The vendors at the Fest were:

Adventure Survivors (Lin & Nan Padgett) — They now have the Power Stones of Ard (Ard I & II) available. The Padgetts produce a bi-monthly newsletter for adventure game lovers, and are a great source for adventure solutions. *24 Perthshire Dr, Peachtree City, GA 30269 (404)487-8461*

Kala Software (Ed Hathaway) — unable to attend due to family emergency. Has UltiMuse music generating software (MIDI) for OS-9 and OSK. *3801 Brown Bark Dr, Greensboro, NC 27410 (910)294-1558*

Delmar Company (Ed Gresick) — unable to attend due to industrial business. Sells the System IV and System V OSK systems and G-Windows for these and OS-9000. *PO Box 78 / Middletown Plaza, Middletown, DE 19709 (302)378-2555*

Northern Xposure (Colin McKay et al) NitroS-9 v1.20, an interim release pending v1.21, was available. Bob van der Poel software and White's Speedisk for OSK were also available. *7 Greenboro Cres, Ottawa, ON / Canada / K1T 1W6 (613)736-0329*

Hawksoft (Chris & Nancy Hawks) — Updates to Sound (now called WAVES) & CD Player (UCDP), both for the MM/1, were available. *244 S Randall Rd, Ste. 172, Elgin, IL 60123 (708)742-3084*

SBUG (Andre Lavelle) — Lot's of computer stuff, of every shape and size! *1251 W Sepulveda Blv #400, Torrence, CA 90502 (310)539-9702*

OS-9 Users Group, Inc. — The booth was manned by Carl Boll and Howard Luckey. A CD-i demo machine was running throughout the fest, and some literature describing industrial OS-9 products was also available. *6158 W 63rd St, Ste 109, Chicago, IL 60638 (312)735-6087*

DALTRUG (Lee Veal) — Planet Engine (OS-9 Astronomy program) was again available, along with Cosmic Jewelry made by Lee's wife. A port of Planet Engine to the MM/1 is in the works. *833 Woodhaven Lane, Grand Prairie, TX 75052 (214)264-3610*

Glenside Color Computer Club — This booth was manned by Brian Schubring & Tony Podraza from GCCC. They were taking memberships to the club that sponsors the Chicago Fest. *119 Adobe Cir, Carpentersville, IL 60110 (708)428-3576*

Digital Frontier Productions (Eric Crichlow) — Gold Runner 2000 for the MM/1 and Digger for OS-9 were available (similar to old CoCo Gold Runner). These first-

timers did good business, and were impressed by the show. All the proceeds from Gold Runner 2000 went towards purchasing an engagement ring for Eric's girlfriend, Stephenie. Good luck, Eric! *2300 W San Angelo #2070, Gilbert, AZ (602)892-2161*

CoNect (Rick Ulland) — unable to attend due to car trouble. See ad in this issue. *449 South 90th St, Milwaukee, WI 53214 (414)258-2989*

FARNA Systems (Frank Swygent) — Frank was selling 268'm subscriptions, along with the new CoCo Family Recorder/OS-9, and KenTon SCSI interfaces. Winner of the biggest banner award! *Box 321, Warner Robins, GA 31099 (912)328-7859*

RC Smith "Gently Used Hardware & Software." *524 Martin Rd, Stone Mountain, GA 30088 (404)469-6601*

Al Dages "Gently Used Hardware & Software." *4894 Candlewood Ln, Stone Mountain, GA 30088 (404)469-5111*

Atlanta Computer Society Club Sales Our Hosts (Thanks, folks!) *PO Box 80694, Atlanta, GA 30366 (404)636-2991 (BBS)*

Wittman Computer Products (Bill Wittman) — The WCP306 Computer Prototype was available for viewing. This machine will be sold by several vendors, under different names. Expected shipping date for these machines is December 94, so write for details. New at the Atlanta Fest were InfoXpress, DeskTamer, LATerm, and the TVP Point-of-Sale system. *873 Johnson Rd, Churchville, NY 14428 (716)494-1506*

Eugene Adams "The Color Computer Philanthropist!" — Eugene was giving away some of the older CoCo2s he has collected, along with some software. *135 Lloyd Dr, Marietta, GA 30066 (404)422-3097*

Mid Iowa & Country CoCo (Terry Simons) — Upgrade Disk Magazine subscriptions with MI&CC memberships were available. Home-Pac OS, the OS-9 Home Manager was also available. *1328 48th St, Des Moines, IA 50311 (515)279-2576*

Rick's Computer Enterprise — Back again with programs for DECB from Sundog Systems, as well as CFDM subscriptions and a variety of DECB software from now defunct CoCoPro! Rick also displayed the new CoCo 3 emulator on his 486 laptop. *PO Box 276, Liberty, KY 42539 (606)787-5783*

JoTA Productions (Tim Johns) - Tim had available his recently completed sound drivers for the Delmar System V. Those of you with Delmar machines should give Tim a call — the driver works great through the built-in

speaker on these machines. Sound blaster support is planned. Some other projects "in the works" should be available in Chicago. *PO Box 4657, Nacogdoches, TX 75962 (409)560-1876*

Klystronics — Written words couldn't do this product justice. They ran out of Klystrons early in the show, but promised to bring more to Chicago. Once again, I missed the chance to meet these people, as they left early on Sunday, and all the business cards on their table were gone. (Address not available.)

Strongware (John Strong) — For the MM/1, SPaint & Graphics Tablet support were available, in addition to their other programs for OS-9/6809. John also remarked that "It was a pleasure meeting the Klystronics rep". *PO Box 361 / Matthews, IN 46957 (317)998-7558*

Sub-Etha Software (Allen Huffman & Terry Todd) — New were Invaders09, and the Blobstop pack — hardware and software modifications to correct the BLOB problems reported on some systems. Updates to other programs, such as Towel, MultiBoot, and the Fest game were also available. Allen also had a photo album covering all the Fests since 1990. Please note that the telephone number printed in the Fest guide is for Allen's Grandmother — please don't use it! *PO Box 152442, Lufkin, TX 75915*

Blackhawk Enterprises (David Graham) — "Home of the MM/1" New 8 Meg upgrade Cards were being shown. (David did note that Klystronics seemed to be generating a LOT of attention.) *PO Box 10552, Enid, OK 73706 (405)234-2347*

Roy Shoaf — "Gently Used Hardware & Software." *293 W Junction Ct, Kennesaw, GA 30144 (404)928-4047*

Color Systems (Zack Sessions) — KChess (a GNU Chess interface for the MM/1), and the X10 Home Controller software were new to Atlanta. With his profits from KChess sales in Chicago, Zack purchased a flatbed color scanner, and was scanning images for all who asked. (Thanks, Zack!) *PO Box 540, Castle Hayne, NC 28429 (910)675-1706*

Nobody (Scott Proctor) — Sale of surplus & spare equipment. Note: Scott runs "Nobody's Home BBS" *PO Box 884, Ocala, FL 34478-0884 (904)245-9396 (904)245-6585 (BBS)*

Ron Bull — Sale of surplus & spare equipment. *115 Ann St, Duncannon, PA 17020-1204 (717)834-4314 (717)834-4071 (BBS)*

Don't forget to write or call vendors to request a catalogue or more information if anything above catches your interest. All the vendors carry many more item's than I've listed here.

The Lectures

This time I abandoned my booth, and managed to attend two of the lectures — the ones given by Kevin Darling on Saturday, and Boisy Pitre on Sunday.

Saturday

1:00- 1:55 "Supporting the CoCo: The Next Step" by Allen Huffman. The main point was to encourage people to RESPOND. Write magazines to let them know what you think, tell vendors if you like their programs (or if you don't!), contribute articles, letters, code, etc. Get involved to keep things going. This is a major problem right now - people leave due to a perceived lack of interest by the community.

2:00- 2:55 "CoCo Hard Drives" by Frank Swygent. Frank talked about the different types of hard drives currently in use and also plugged the Ken-Ton units he is now selling (the only ones still being made). Also discussed cost of adding a hard drive to CoCo.

3:00- 3:55 "Open OS-9 Forum" by Kevin Darling. Those who attended were updated on Kevin's current career writing software for video gambling systems. Look for an article in an upcoming issue of the OS-9 Users Group newsletter (MOTD) and a shortened version 268'm about these machines.

Sunday

10:00-10:55 "The Future of Desktop OS-9" by David Graham. David discussed a variety of topics during this seminar, enough that I gave it a separate section below.

11:00-11:55 "Graphics and Game Programming Forum" by John Strong. Once again John outlined how to create a game, from early concept to final product.

12:00-12:55 "Inside CD-i" by Boisy Pitre. A fascinating demo of the capabilities of the CD-i machines. Boisy also demonstrated his bootable CD-i disk which gives shell access on a CD-i machine via a terminal. The videogame "Dragon-slayer", with the exploits of Dirk Daring, was one of the recent titles released. It really is an interactive cartoon/adventure! (And one of my all-time favourites.)

The Future of Desktop OS-9

To begin his seminar, David Graham defined a new term: Desktop OS-9. This means any Personal computing or host development use of OS-9. (Author's note: While David is referring to OS-9 in general, OS-9/68000 was the primary emphasis in his seminar.) He did note that the past history has been rather dismal, but things are looking up. Four new terminal programs have appeared over the last year, and one new terminal program and a major database (FoxBase compatible) will be coming this year. Some very nice games are out now — notably Gold Runner 2000 by Eric Crichton. A lot of projects are also in the works, including a CD

File manager and other multimedia programs to be completed during next year or two.

Program capability is increasing as people become more familiar with OS-9/68000 and Desktop vendors are banding together to assure more software of higher quality in more timely fashion. The Desktop OS-9 Developers and Marketing Association (unofficial name) is being formed to provide formal support for these activities. David noted that the MM/1 will return to full hardware production within 1 month, with new backplane available at the Fest.

A new machine, the MM/1B based on 68306 board by Carl Kreider will be available late 1994/early 1995. This is the same machine described above under Wittman Computer Products as the WCP306 (see the November issue, "Nine Flavors of OS-9; AT306" for details). Essentially it is a 16MHz 68306 board with six PC/AT 16 bit slots and connectors for two IDE drives and two floppy drives. A VGA/SVGA card is used for graphics. OS-9/68000 v3.0 is included, along with the MGR windowing package. Price for the motherboard and software is expected to be in the \$400.00 range — a fantastic price!

Following the same theme as Allen Huffman mentioned in his seminar, David also emphasized the need for better communication between users and authors/vendors. Users must ask for they want, and pay for it when they get it. There have been too many episodes of users saying they want something, and not buying it when vendor does supply it (Tape drives, CD-ROM drives example) - and vendors must supply better quality merchandise.

David emphasized the importance of gaining new users and making inroads into other markets by accelerating development and becoming more professional. Cultivating real time and embedded systems people is a must. Look for big things in next 1 to 2 years.

Chicago 95

Not related to Windows 95, the Fourth Annual 'Last' Chicago CoCoFest is scheduled for 29-30 April 1995. Be there, or be a right-angled equilateral quadrilateral!

Colin McKay
Northern Xposure
cmckay@northx.isis.org

(Palindrome... beta version)
(Think about it a while!)

< 268'm >

Announcing the

4th Annual "Last" Chicago CoCoFest

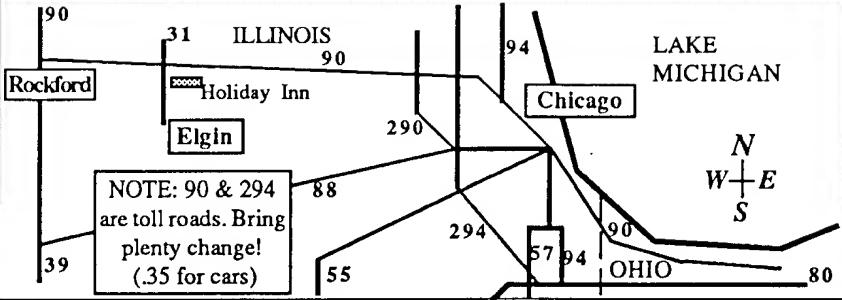
April 29 & 30, 1995

at the

Elgin Holiday Inn

a Holidome Indoor Recreation Center

345 West River Road, Elgin, IL



SMALL GRAFX ETC.

"Y" & "TRI" cables. Special 40 pin male/female end connectors, priced EACH CONNECTOR ----- \$6.50
Rainbow 40 wire ribbon cable, per foot ----- \$1.00
Hitachi 63C09E CPU and Socket ----- \$13.00
512K Upgrades, with RAM chips ----- \$72.00
MPI Upgrades
 For all large MPIs (PAL chip) ----- \$10.00
 For small #26-3124 MPI (satellite board) ----- \$10.00
Serial to Parallel Convertor with 64K buffer, cables, and external power supply ----- \$50.00
2400 baud Hayes compatible external modems ----- \$40.00

ADD \$2.00 S&H TO EACH ORDER

SERVICE, PARTS, & HARD TO FIND SOFTWARE WITH COMPLETE DOCUMENTATION AVAILABLE. INKS & REFILL KITS FOR CGP-220, CANON, & HP INK-JET PRINTERS, RIBBONS & Ver. 6 EPROM FOR CGP-220 PRINTER (BOLD MODE), CUSTOM COLOR PRINTING.

**TERRY LARAWAY, 41 N.W. DONSEE DRIVE
BREMERTON, WA 98310 206-692-5374**

Show Hours:

Sat., April 29 10:00AM-6:00PM
Sun., April 30 10:00AM-4:00PM

Admission:

\$5.00 in advance, \$8.00 at door
(2 day pass only, order before 16 April)

Address for advance tickets:

Tony Podraza, Fest Coordinator
119 Adobe Circle
Carpentersville, IL 60110-1101

Reservations:

1-708-695-5000
1-800-465-4329

Sponsored by:

Glenside CoCo Club

Vendor Information:

Booth Price: \$35 (\$30 for 2nd)
Member Price: \$30
(Membership can be purchased at registration -- \$15 per year)
Reservations must be received no later than 3/25/95. Deposit of \$20 per booth required with balance due no later than 4/26/95. Balance received after 4/16/95 subject to a 20% late fee. Vendor setup Sat, April 29 5:30AM-9:45AM

DON'T FORGET TO RENEW YOUR 268'm SUBSCRIPTION EARLY!

As of 01 Jan 95, rates will increase to \$25 for domestic mail (no change for Canada and Mexico). We will accept renewals for the next year (one year only) from current subscribers until 01 February 1995. Sorry for the inconvenience, but we do want to continue offering you the quality support you deserve!

BASIC in Color

BASIC Support for the CoCo!

Fred Remin

Many of you still use and enjoy Tandy's Extended Color BASIC and Disk Extended Color BASIC (I'll use DECB to refer to both from now on). Several readers have asked for more DECB articles, so I've decided to start a column specifically for BASIC users.

There is one thing I want to get clear right from the start. This is YOUR support column! If you don't write, call, or send me e-mail, I won't have anything to put here. I will devote space in each issue exclusively to DECB, but don't make me print a blank page because nothing has been sent in!

NO topic is too simple. Even tips on how best to use a tape recorder would help someone. And don't think you shouldn't cover something that has been covered pretty well in past magazines. There are many out there who are just starting with their handed down CoCo, so there are no subjects that "everyone knows"! Remember the one and two liners from Rainbow? Little things like that will make interesting fillers throughout the magazine.

I need your input to keep this column as a regular feature. I also need a better name. Let me know what you think would be a better name for a DECB only column.

Since I really have nothing else to cover in this issue, I'm going to start with a beginners tutorial. This tutorial is reprinted with permission from Australia's only CoCo magazine--*CoCo-Link*. This will run for the next one or two issues, but don't let that keep you from sending in submissions now... I need time to plan each issue's contents!

I'm going to start this new Basic support column with a beginners tutorial. This tutorial is based on a series printed in Australia's CoCo-Link magazine (written by Fred Remin). My apologies to Fred for all the editing and adding! (There was nothing wrong with the original articles, I just wanted to add and rearrange some of the info.)

What is a CoCo?

"CoCo" is the affectionate name given to the Tandy Color Computer and encompasses the original Color Computer (usually referred to as the CoCo 1), the Color Computer 2, and the Color Computer 3. This name was unofficially given to the Color Computer by Dave Lagerquist (editor/publisher of the late "Chromasette" magazine on tape) in October 1980. Tandy soon followed suit by referring to the machine as the CoCo in several of their publications and later manuals.

The original CoCo was grey in color. It contained Color BASIC (no extended) in 8K of ROM and had 4K of RAM, all for \$399.00. It was expandable to 16K of ROM (Extended Color BASIC) and 16K of RAM.

The next CoCo was the CoCo 2. It had a white case which was shorter (front to back) than the original. It came with 8K ROM and 16K RAM and was upgradeable to 16K/64K, respectively.

The CoCo 3 finally debuted in 1986, six years after the first. With this model we finally got an 80 column text display, high resolution color graphics, and more memory! The CoCo 3 came with 16K ROM (extended BASIC was standard) and 128K of RAM. Tandy also offered an upgrade to 512K (also offered by other vendors). Disto even offered a 1MB and then a 2MB upgrade, though these aren't very useful for BASIC programming.

How much memory do I have?

One question often asked of me is "how can I tell how much memory my system has?". A quick way of doing this, once booting (turning on) your system, is to type in the following:
PRINT MEM <enter>

The result shown on the screen will give you a ROUGH indication of the amount of RAM that you have, e.g.:

Around 4,000 = 8K RAM

Around 8,000 = 16K RAM

Around 16-18,000 = 32K RAM

Around 22-28,000 = 64K RAM+

Even though you have a 128K or 512K CoCo 3, it will still only give a result between 22-28,000. This is because a copy of the CoCo 2 ROM is used. All CoCo 3s came with 128K RAM from the factory. Look inside the system through the slots at the top left corner. You will see either some empty white connectors or a small circuit board close to the top of the machine. If the circuit board is present you have a 512K machine. If a Disto 1MB or 2MB upgrade is present, there will also be a circuit board near the top of the machine visible through the slots on the right.

Setting up your system

The first thing to do is to determine where you are going to have the system set up. For example, when I first started I used the family TV but this became a real nuisance when the kids wanted to watch TV, so now I have a separate area for my system. Some of the things you should consider are:

a. Power points (outlets). How many power points will you need for your system? Take into account the computer, tape and/or disk drive, TV/monitor, printer, multipak, and any other peripherals that you may have or need in the future. It is generally a good idea to get a six outlet power strip. Then the entire system can be turned on and off at once, and this is usually enough outlets.

b. Distractions. Is the system in the main path between the front door and bedrooms or in full view of the TV? It is best placed where there are a minimum of distractions so that you can get on with enjoying your computer.

c. Ventilation. As your system is operating it is generating heat. Make sure there is adequate ventilation around the computer. If it is stuck in a tight nook on a computer desk, you may need a small desktop fan blowing across it and the disk drives to provide adequate cooling. Except in extreme

cases, the CoCo will stay cool on an open desktop without additional cooling. Intermittent errors for apparently no reason are usually caused by lack of cooling, especially if the errors only occur after 10-15 minutes of operation. The older models are more susceptible to heat than the 3.

d. Space. Is the space you have selected going to allow you to expand? Is there room for all you need, and maybe that printer you have yet to get? You also need room for books, disk, tapes, and magazines. You want to keep a copy of "68' micros" handy!

e. Phone! Is the system close to a phone? This will come in handy for when you have a problem and you want to ring someone for help. It will also come in handy when you get a modem added to your system for access to the "Information Highway".

f. Lighting. Unless you want to damage your eyes, I would suggest that you have adequate lighting. At the same time, ensure that any light is not directly on your screen. This can be a real distraction can make it hard for you to properly see the screen.

By taking the above into consideration, and a few more you may think of yourself, you will ensure that you have many enjoyable hours at your computer.

Media handling

Whether you use tapes or disks you MUST be aware of any magnetic field. DO NOT place your tapes or disks directly next to or on top of any electronic device. The magnetic fields generated could cause some of the data on the data to be destroyed. Most computer equipment is usually shielded enough to prevent damage, but don't take any chances! If you are using a TV, stay clear of the left side (facing TV, your left). This is where the flyback transformer is in most TV sets. That transformer creates quite a magnetic field and wasn't shielded to be around disks and tapes!

Before you turn it on

The next step is to READ THE IN-

STRUCTIONS that came with your system ("Introducing Your Color Computer") on how to connect everything. All the ports on the back of the CoCo are clearly marked, so if you don't have the manual this is generally no problem. One thing that you MUST remember, however, is to NEVER, EVER remove or insert a cartridge, multipak, or disk drive controller in the side of the CoCo with the power on! This could cause the 6809 processor to blow. Most "dead" CoCos are caused by this simple mistake. Don't bump or wiggle anything in the slot after the power is on. This isn't as important if you are using a multipak, as the multipak has buffers to prevent damage to the computer. You could, however, damage the cartridge being inserted.

Memory and BASIC version

The amount of memory available to you I have already discussed above. This will determine what programs will or will not work on your machine. For example, a game which requires 128K to operate will not work on a 64K CoCo. Any program that works on a CoCo 1 or 2 will usually work on a CoCo 3, but not the other way around.

Another problem area with the CoCo 1 and 2 is the version of BASIC installed on your machine. A program which requires "Extended Color BASIC" will not work on a "Color BASIC" machine. The version of BASIC will appear on the first line of your screen when the computer is first turned on without a game cartridge. If a disk controller is in the cartridge slot, the word "Disk" will be added to the message. CoCo1s and 2s will have BASIC version 1 while CoCo3s will have version 2. Version 2 will generally run all version 1 programs.

If your computer displays something other than a BASIC version, you have a modified ROM. The most popular modification is to add something called ADOS. Since even ADOS is customizable, you need to get in touch with the computer's previous owner if possible to determine what options

were installed. If you can't reach the owner, write and I'll try to help.

Tape versus disk storage

You should be aware that the CoCo can use tape, disk, or both as a means of information storage and retrieval. If a disk system is attached to your CoCo you will lose about 4K of memory (this is used by disk BASIC). Because of this some tape programs will no longer run. Some can be "offset loaded" with a disk to tape utility and be made to run from disk, while some cannot.

The "advantages" to using tape is that it is cheap and works with ROM paks. The disadvantage is that it isn't always reliable and it is SLOW. Reliability improves with tape quality, so always use at least mid-grade tapes of the shortest length available. NEVER try to use the cheap "3 for \$2" audio tapes! ALWAYS make at least two saves of any program or data with some blank tape in between saves, just in case!

Disk drives are really convenient if you intend to use your computer a lot. In the old days, there was good reason to use tape instead of disk, as a disk system cost much more than the computer (in 1983, an Extended BASIC CoCo 2 was \$319.95 and a single disk drive and controller was \$499!). Now they are about the same, with used single disk system going for \$5.00-\$75.00. For the best games and application programs, disk drives are a must! So if you don't have a disk system do be on the look-out for one. If you can't find a used one in your area, send an SASE to "68' micros" and I'll see what I can find for you.

Next time...

We have covered the basics of setting up the CoCo in this article. Next time we will get a little deeper into the actual operation of the machine. If you need any equipment let me know. Save the operation questions for the next issue!

< 268'm >

G-WINDOWS

Last month I presented an overview of G-Windows and its origin. This month I'll present my usage of G-Windows for my business. I may be at the computer 12 or more hours a day, day after day. I don't want to waste time waiting for the computer or searching for information. Comfort is paramount. Most of the time, I can't just stop working and take a break or relax. And I can't afford eye strain - these old eyes just can't take it.

Since I spend considerable time on the phone, I've switched to a cordless, headset phone leaving my hands free for other activities and giving me mobility. The point of all this is simply that I won't be a slave to machines be they computers, phones or whatever. Rather, they are to serve me.

I have four machines running G-Windows. A SYSTEM IV (68000/16 MHz) - my 'business' machine, a second SYSTEM IV which is also equipped with an ALT86 card for running MS-DOS and used as a demo, a SYSTEM V (68020/33 MHz but sometimes at 25 MHz) and a 486DX2@66/33 MHz running OS-9000. The SYSTEM V and 486 are used for programming and as demos.

I have a third SYSTEM IV used to run my retail store. The later system is not equipped with a VGA card or keyboard hence cannot run G-Windows - it is configured as a serial system and has 8 serial ports. I still run my CoCo 3 but only for UUCP (Mark Griffith's version) but it is destined to be retired soon when I start using the UUCP port to OSK by Bob Billson and Boisy Pitre.

Recently, my budget permitted me to acquire a 17 inch monitor. Previously, I used 14 inch VGA monitors. While the computers and 14 inch monitors supported 1024 x 768 screens, I used the 800 x 600 mode most of the time when dealing with text - the fonts were too small in the 1024 x 768 mode on a 14 inch monitor for me use comfortably. With the change in monitor, I now use the 1024 x 768 mode almost exclusively; the fonts are easily readable. I can have 4 full-sized (80 x 24) windows fully visible using the normal fonts in the 1024 x 768 mode (G-Windows includes 3 sizes of fonts which are user selectable when opening a new window.

The sizes are small - 5 x 10 pixels, normal - 6 x 13 pixels and large - 8 x 16 pixels.) Rarely is all the information I require in a single file nor could it be displayed on one 80 x 24 windows if it were. Having four full sized windows viewable simultaneously permits access to the information I require. Sure, I could (and did) flip windows but time is lost and the chance for errors increases. Or, I could use SQL to assemble the information but unless I need this information for many items, customers, vendors, or whatever, it isn't worth the time to set-up and run SQL.

None of the software I use regularly is designed for G-Windows. The programs are all designed to run with text terminals. The main ones are SCULPTOR QuickEd, DataDex, SC, MW's K&R C Compiler and separately, their assembler and linker and Ultra C. Typically, I'll have separate windows running several SCULPTOR programs; i.e., the SCULPTOR main menu, inventory, inventory history, purchase orders, customer file, vendor file and, before the end of the day, additional programs called from the SCULPTOR menu. I have a separate customer list running under DataDex. The above programs are starting from a shell script file which is executed when G-Windows is started. Other programs are started using the G-Windows 'custom' menu, point and click using the file recognizer (more on these later) or from the command line in a shell window.

I mentioned that I use 2 database programs; SCULPTOR and DataDex. The software I use for the business is written under SCULPTOR. SCULPTOR, however, like all relational databases, does not lend itself to variable length fields and records. DataDex was written to support free form records of variable size. The two work well together - especially under G-Windows.

To give an example, when a customer calls in, I like to call up his file in SCULPTOR. Often this does not provide a complete record of his interests, problems, etc. For this, I use DataDex. After I've called a customer's record up in SCULPTOR I can press Shift-F9 to get the customer's DataDex file. If it exists, the record is displayed and I can review and/or add to it. If it doesn't, the

file is displayed and I can easily add a new customer record to it. If the customer doesn't exist in the SCULPTOR file, an error will be returned and I can immediately switch to the DataDex customer file and enter customer data.

I can do everything I've described from a non-G-Windows display but I can't view files simultaneously. I could export data from the DataDex file to an ASCII file (or pipe) to be imported to the SCULPTOR customer list. This requires that information be input in a prescribed sequence. I find it best to input the information from the customer as he gives it to me and it may (and will) be in any order. Using the 'copy and paste' capabilities of G-Windows I can transfer the appropriate information to the SCULPTOR customer file. I can do similar things with parts pricing and other information that I obtain over the phone. If the information is already in electronic form, using 'copy and paste' I can copy information from any file to any file I wish. There is no need to re-enter the data manually.

I can't say programming under G-Windows suddenly becomes a joy but it is much less tedious than using a simple terminal particularly when a high resolution display is used. The editor can be run in one window. Header files, defines, declarations, other source code and other sections of the same source code can be displayed in other windows. If the windows are full sized, reference to and use of their contents makes the process of writing programs easier and me less prone to introducing errors. And the debugging process is made easier. I can have the editor running in one window. When I'm

For comments or questions, Ed can be reached via this magazine or:

E-mail:
EDELMAR@delphi.com
76576,3312@CompuServe.com

U.S. Mail:
PO Box 78
Middletown, DE 19709

Telephone:
302-378-2555 Voice
302-378-2556 FAX

finished with my changes, the file is saved. I can switch to another window and run lint, the makefile or the compiler or assembler. Errors can be quickly corrected and the process repeated. When I finally have a program I can run, it can be tested. Since I often obtain results not planned for (undocumented features) there is an edit, compile and test cycle to go through. G-Windows includes a history of the last 50 commands. Using this capability can reduce some of the tedium of this process. After making and saving the changes, I switch to a shell window and using the up-arrow key, select the last compile command with only a couple of key presses. When compiled, a couple key presses provide the command line last used to run the program.

As can be seen, G-Windows introduces an environment or 'layer' between the operating system and applications. The power of this environment becomes more apparent and useful as the capability of the display system is improved. The number of processes which may be run are not limited by G-Windows but by memory/operating system considerations.

Until switching to a larger monitor and using it regularly, I didn't fully appreciate comparisons others had made with Sun workstations and other workstation of that class. Speed of performance of G-Windows (on a SYSTEM V) compared favorably. But, they said the feel wasn't quit the same. While these workstations can be obtained with 17" monitors, they're normally equipped with 20" monitors and usually run with a resolution of 1280 x 1024 or better. Since switching to the larger monitor, two workstation users have worked with my SYSTEM V and have commented they felt completely comfortable—that the differences between G-Windows and X-Windows (on the Suns) they use were trivial. One did allow that he still prefers a 20" monitor. I should point out that both of these individuals are expert OS-9 programmers and normally program for Motorola and Force VME systems. They use Sun (SPARC) work stations (UNIX) and use MW's FAST TRACK for programming. In the next issue, I'll discuss some of the industries where G-Windows is being used.

What do you mean you have a friend interested in the CoCo (or OS-9, or OSK!) and they don't have a subscription?

Trek!

Animated graphics in DECB

Pete Blackwell

Here's a neat little animated graphic routine. Look over the routine and see how Pete manages to make the "starship" move across the background. Once you figure that out, try to find a way to make the starship pass in front of the last two planets instead of behind them!

```

2 ' Pete Blackwell
4 '1408 1/2 McFadden St.
5 'Paris TN 38242
6 '(901) 644-2635
8 'Nov. 1994
10 WIDTH40:POKE 65497,0
20 ON BRK GOTO 360
30 HBUFF 1,100: HBUFF 2,100: HBUFF
3,100
40 HSCREEN2
50 PALETTE0,0:PALETTE1,3: PALE
TTE2,63
60 HDRAW"BM3,3C4FR6F2E1L8U4"
70 HDRAW"BM0,1C5FR7F3E2L9U5"
80 HGET(0,0)-(15,7),1: HLINE(0,0)-(15,7),PRESET,BF
90 FORS=1TO20:HSET(RND(320),
RND(191), RND(8)):NEXTS
100 HDRAW"BM130,30 ;S2C7D2L3D2L2D
3R2D2L2D2L1D1H3D2H1L2U1L1U2H2U1
L2H2L2H2L2H2D2L2D2L2H2L2D2L2U3H
2U3H1U2H1U1H1R2E3ER3H3R3U3R3H3
R2H2R3D2R2D2R3G2R3G2R3D2R2U3R3
U2R2U3R2U3R3U2R3U2R3G2R3G2R3
R2G3R3G3R2G3R3F3R3R2R2F3R2F2R1
F3D2H2D3H2D3 H2L2D2L2H2D2L2H2"
120 HDRAW"BM130,30;S3C7D2L3D2L2D
3R2D2L2D2L1D1H3D2 H1L2U1L1U2H2U1
L2H2L2H2L2H2D2L2D2L2H2L2D2L2U3H2
U3H1U2H1U1H1R2E3ER3H3R3U3R3H3R2
H2R3D2R2D2R3G2R3G2R3D2R2U3R3U2R
2U3R2U3R3U2R3U2R3G2R3G2R3G3R2G
3R3G3R2G3R3F3R3F2R3F2R1F3D2H
2D3 H2D3H2 2L2D2L2H2D2L2H2"
140 HDRAW"BM130,30;S5C7D2L3D2L2D
3R2D2L2D2L1D1H3D2H1L2U1L1U2H2U1
L2H2L2H2L2H2D2L2D2L2H2 L2D2L2U3H2
U3H1U2H1U1H1R2E3ER3H3R3U3R3H3R
2H2R3D2R2D2R3G2R3G2R3D2R2U3R3U
2R2U3R2U3R3U2R3U2R3G2R3G2R3G3R
2G3R3G3R2G3R3F3R3F2R3F3R2F2R1F3
D2H2D3H2 D3H2L2D2L2H2D2L2H2"
145 HDRAW"BM130,30;S7C7D2L3D2L2D
3R2D2L2D2L1D1H3D2H1L2U1L1U2H2U1
L2H2L2H2L2H2D2L2D2L2H2L2D2L2U3H
2U3H1U2H1U1H1R2E3ER3H3R3U3R3H3R
R2H2R3D2R2D2R3G2R3G2R3D2 R2U3R
3U2R2U3R2U3R3U2R3U2R3G2R3G2R3
G3R2G3R3G3R2G3R3F3R3F2R3F3R2F2
R1F3D2H2D3H2D3 H2L2D2L2H2D2L2H2"
147 HPAINT(126,29),4,7: HPAINT(120,24),
7,7: HPAINT(127,27),3,7: HPAINT(110,25),
1,7: HPAINT(88,25),3,7
148 HPAINT(120,34),5,7: HPAINT(110,38)
,7,7: HPAINT(134,26),9,7
149 HDRAW"BM145,54;C7L2U2L1H2L1D2
L2U1H2D1L1U1H1L2H2L1D2": HPAINT(143,
50),3,7: HDRAW"BM95,15;C7D2R1D2 L2
D1 R1D1R1D2F2D1L1D1": HPAINT(94,18),
0,7
150 HDRAW"BM216,23;C6E3;BM216,23;
F3;BM216,23;G3;BM216,23; H3; BM216,23;
L3;BM216,23; R3; BM216, 23; U3; BM216,
23; D3": HCIRCLE (216, 23),2,4
151 HCIRCLE(110,25),2,5: HDRAW"BM110,
25; C5E1; BM110,25; F1; BM110,25; G1; BM
110,25; H1": HCIRCLE(115,15),2,7: HDRAW
"BM 115, 15;C7 E2;BM115, 15; F2; BM 115,
15;G 2;BM115, 15;H2; BM115, 15;
L2R4;BM 115,15;U2D4"
152 HDRAW"BM120,165;C4E2; BM 120,
165;G2;B M120,165;F2;BM120,165;H2; BM
12 0,165;U2; BM120, 165; D2; BM120, 165;
L2;BM120,165;R2": H CIRCLE(120,165),2,1
160 HCIRCLE(70,85),25,1: HPAINT(72,83),
1,1
170 HCIRCLE (71,85),24,7,1,,75,,22: HCIRC
LE(69,85),24,7,1,,75,,22: HCIRCLE(70,85),
24,7,1,,75,,23
180 HCIRCLE(58,80),3,8,1: HPAINT (59,79)
,8,8
190 HCIRCLE(160,96),30,1: HPAINT(160,
96), 6,1
200 HCIRCLE(159,96),29,7,1,,75,,24: HCIRC
LE(160,96),29,7,1,,75,,24: HCIRCLE(161,
96),29,7,1,,75,,25
210 HCIRCLE(250,105),40,7: HPAINT(255,
102),7,7
220 HCIRCLE(251,105),38,1,1,,75,,22
230 HDRAW"BM75,70;S4C8G4L6D2 R3D3
F4D3H4L2L2H2L2F3G4D2L2D2L2H3D3R3
D2L2D3R3D3R3D2R3D2R2D1R4U4E4D3
R2D2R3U4R2U3R2U2R2H5D3L2U3L1U3
R2U2R3U2E2H2L2H2 UH1"
240 HPAINT(70,75),8,8
250 HDRAW"BM157,70;C8G2L3G2L3G2
D3R2D3F4D3H2L2H3L2F3D2L2D2L2H2
D3R3D2L2D3R3D3R3D2R2U2R2U2R3E4
R3E2R3E2U2R2F3R3U2R2U3L2U3L2U2E
3L2U3L3U2H3L3H2"
260 HPAINT(158,90),8,8
270 FORS1=1TO200:HSET(RND (320),
RND (191), RND(8)):NEXTS1
280 FOR X=0 TO 320 STEP4
290 HGET(X,100)-(X+15,107),2
300 HPUT(X,100)-(X+15,107),1,OR
310 'PLAY"V31L1T255O1C
320 FOR T=1 TO 20:NEXTT
330 HPUT(X,100)-(X+15,107),2,PSET
340 NEXTX
350 GOTO280
360 WIDTH 32

```

< 268'm >

The Hardware Hacker

More Tips on Fixing Color Computers and Accessories; Convert DS69A to B

Dr. Marty Goodman

With support from Tandy for the Color Computer long gone, you can't always take a dead CoCo or disk controller to Tandy Repair and expect to get it fixed any more. The same is true for most if not all third party accessories. Thus, I'm going to devote this column to tips on how a competent hardware hacker can better go about repairing CoCo equipment. I'll include experiences of mine and others on specifically what to look for to maximize the chance you can make a repair. I'm also going to include some preliminary information on modifying the Microworks DS69A video digitizer to turn it into a DS69B that works with the Color Computer 3.

Fixing a Color Computer 3:

When a CoCo 3 dies with just a blank green screen showing, one should first try a few of the easy things: Try removing, then re-seating, the GIME chip. Care must be taken in doing this. Either use a proper, fancy, PLCC chip removal tool, or be VERY careful in the use of jeweler's screwdrivers to lever the chip up. Try re-seating the 512K board, if the computer has one in it. Often a 512K board jammed too far in or too loosely in is the source of the problem.

Other easy-to-test causes of failure would involve a bad memory chip (replace memory chips... that's not hard or expensive to do), or oxidized contacts on the edge connectors in the system. Clean all edge connectors in the entire system, Multipak and all cards plugged into it with a pencil eraser. Clean them on both sides. This means the cases will have to come apart. If all this fails, you have to get more serious.

The single most common thing to cause any CoCo to die with a blank green screen visible is the fried 6809E cpu chip. This, in turn, is often caused by wiggling a cartridge as one inserts or removes it from a Multipak plugged in to the CoCo with the Multipak power on (it matters not whether or not the CoCo's power is on at the time). What has happened here is that the + and - 12 volts lines get shorted to their adjacent *HALT and *NMI lines, sending a blast of nasty

voltage right into the 68B09E chip. The fix consists of removing the 6809E from the CoCo mother board, installing a socket where it used to be, and plugging in to that socket a new, good, 6809E chip. Chip removal is most easily accomplished "destructively": One cuts each of the 40 pins of the chip, then lifts out the chip. Then one uses a needle-nosed pliers and a soldering iron to heat each pad and remove each remnant of the pins of the chip from its hole. Then one uses a solder sucker to clean out the pads, so one can install a socket.

I'd guess that roughly 4 out of 5 CoCo's that die or are acquired dead with a blank green screen can be fixed simply by replacing the cpu chip. It's most fortunate that it's quite rare for the (irreplaceable) GIME chip to get blown.

If you have an extension keyboard, or are messing with the keyboard connector, it's possible to zap the keyboard PIA chip. I did this once or twice when I was developing my keyboard extender cable. I've been told that an ordinary 68B21 PIA chip will substitute for the keyboard PIA chip. The Tandy keyboard PIA actually has some capacitors built into it supposedly to help with keybounce. A standard PIA causes no noticeable increase in bounce, however.

If your CoCo was working, then dies with a totally BLANK screen, listen carefully when you turn the power to it on. IF you hear the distinctive DOUBLE click of the cassette relay at power up, you have a problem in the video circuitry. IF you hear no or a single click, one possibility is a broken main clock crystal. This is especially a possibility if you have DROPPED the Color Computer. Use an oscilloscope to check to see if the crystal is oscillating. If not, replace it with another 8X colorburst crystal.

If you check all these things are still unable to fix the CoCo, you're into more serious repair. The next step usually involves putting an oscilloscope on all the lines of chip after chip, looking at and comparing signals. For example, look at the address, data, *CAS, and *RAS signals going to each DRAM chip. In one case, after WEEKS spent by a very

experienced repair tech (George Bethea of New York State) trying to fix a CoCo 3 that had worked for three years then had suddenly died, this approach showed that there was NO *CAS signal going to the rear-most of the four 4 X 64K DRAMs in a dead 128K CoCo 3. Note that this was AFTER this fellow had non-destructively removed and tested the CPU, the ROM, the GIME, the PIA, and the memory chips in the machine. After he noticed the absence of the *CAS signal, he used a continuity tester to discover that the trace between the third and fourth DRAM chip carrying the *CAS line had corroded and disappeared at a spot that was concealed UNDERNEATH the third DRAM socket, just where it left the pad of the third DRAM to become a trace going to the fourth DRAM. Presumably this had been a defective trace to begin with, and over the years had corroded and eroded further. Note that checking address and data and other control lines with an oscilloscope is helpful if you had a previously working CoCo 3 that now is dead after you attempted to socket its 68B09E chip in order to replace it with a 63B09E. Note, too, that traces can leave pins both on top and on bottom of the circuit board, and so can tear during replacement operations on EITHER side.

Exactly the OPPOSITE of this can happen too: I once had a CoCo I whose cassette port ceased to operate. Using a signal tracer (Radio Shack \$10 speaker /amplifier box) it was discovered that the outgoing cassette signal just disappeared... was shorted to ground... at one point in the circuit. Careful examination of the circuit board with a magnifying glass at that area showed a ROGUE TRACE that had literally GROWN between a pad carrying that signal and a very near by ground area on the circuit board. This rogue trace was only a millimeter or so long... but it was enough to short out the cassette line. These are actual examples of rather odd, very hard to fix problems. But a determined technician can often find them. It DOES take a lot of time, and a lot of more obvious things have to be checked for first.

Disk Controllers:

The single most common cause of a dead disk controller is a blown 7406 (or 7416) chip on the *NMI and *HALT lines. When this happens, the *HALT and/or *NMI lines are forced true at all times. This in turn causes the CoCo to power up with a blank green screen whenever you try to power it up with such a blown controller in place in the CoCo or in the Multipak that is connected to the CoCo. This failure is caused by the same event (wiggled cartridge in the Multipak) that blows cpu chips in the main CoCo.

Socket the 7406 or 7416 chip, and replace it with a new 7406. 7416's should be replaced with 7406's, for they are exact replacements for the 7416 and a bit more rugged (rated to higher voltages). Most, if not all, CoCo disk controllers have TWO 7406 (or 7416) chips. Just find the one that is part of the *HALT and *NMI circuitry, and replace that one.

Sometimes other small scale logic chips (74LS00, 74LS02, etc.) associated with the *HALT and *NMI circuitry also get blown. These may have to be replaced, too.

Rick Ulland has reported to me that he's seen disk controllers that had been dropped and suffered from a broken 16 MHz crystal or TTL oscillator. This is easy to check for... just use an oscilloscope on the clock circuitry when power is applied to the controller, looking for oscillations. If the clock is dead, replace the crystal or oscillator module (whichever your particular controller happens to have) with an appropriate replacement part.

I recently found a disk controller that did not crash the computer when connected, but which refused to properly turn on the drive select lights. If I typed POKE &HFF40,1 the drive 0 light did not come on. Similarly, if I typed POKE &HFF40,2 the drive 1 light did not come one. Yet, when I attempted a DIR command, the head of the drive did try to seek tracks. I reasoned here that the register at \$FF40 which controls drive selection was damaged. This register physically IS a 74LS273 chip on almost all models of disk controller (and was on the Tandy FD500 that was I was working on). When I replaced just that one chip

(the LS273 chip) the problem went away, and the controller started functioning.

If you plug in the disk controller and power up and your CoCo comes up with EXTENDED BASIC (Disk ROM not recognized), you may have a blown Disk ROM. Replace it with a new one (it's good to keep spares) and see if this fixes the problem. The controller that had the bad LS273 chip also had a blown Disk Basic ROM chip.

Happily, it's rare for the WD1773 disk controller chip found in the later model Color Computer disk controllers to be blown. That chip is nearly impossible to get hold of these days, though Rick Ulland and SBUG should have a few.

RS232 paks:

The most common problem with RS232 paks is a blown level converter chip. This is typically the 1488 or 1489 chip. Sometimes the 6551 UART is blown, but this is rare. The LS TTL chips in it virtually NEVER get zapped. If they are zapped, probably every other chip on the thing is zapped, too. Of course, some variant RS232 devices use the MAX 232 chip as a level converter. On those, you want to try replacing that chip.

Testing the RS232 pak is facilitated by using a RS232 tester... a thingie with LEDs that show the status of each of the serial port lines. By booting and then trying to operate a term program while watching the lights, you can often get an idea of where to start looking for problems. Note that a plain old external modem can in a pinch often serve as a RS232 line checker, by using the lights ON the modem.

The DS69A and DS69B:

I recently was asked by Terry Simmons if I could help him convert two DS69A's into DS69B's. These were video digitizers made by Microworks. Microworks is no longer in the CoCo market, and told Terry they did not care to support that product. However, they did say that the main difference between the DS69A and DS69B was the programming of the 16L8 address-decoder PAL chip on the unit.

Terry sent me a DS69B and two DS69A units. I desoldered the 16L8 PAL chips

from all of them, socketed all of them, and burned new 16L8 PAL chips for the DS69A's based on the information I read out of the PAL chip on the DS69B. I also checked the 16R8 PAL chip on the digitizers, and found they were all the same. I further noted that the circuit boards for the DS69A and DS69B were identical... both were labelled "DS69A". The 74HC14 on the DS69A had pin 1 sent to ground via a 220 pf NPO type disk capacitor, but the same pin of the same chip on the DS69B was sent to ground via 150 and 120 pf capacitors in parallel. So I added a 50 pf cap in parallel with the 220 pf cap on the DS69A at that spot.

When I returned the modified DS69A's to Terry, he found one of them worked perfectly as a DS69B with the driver software for the DS69B. The other one, he told me, worked, but the image had some funny stripes on it. Terry later informed me that he saw some other small differences between the DS69A and DS69B right at the video input jack on the DS69A. Apparently some resistors and capacitors were different, or added, or subtracted. Thus, if you are trying to do this same conversion, look carefully there and make whatever changes are needed.

For those with PAL programmers, the fuse map for the 16L8 PAL chip of the DS69B digitizer, as read by my ancient device programmer in my lab, is on page 18 of this issue. Note that there exists software to convert a 16L8 fusemap into one that can be burned into a 16V8 GAL chip. 16V8 GALs are cheaper and more easily available than the older technology 16L8 PAL chips.

Readers may contact me if they wish to buy 16L8 PAL chips burned with DS69B addressing fuse map data. Price is \$12 per chip, shipping included. Same is true for Tandy 26-3024 Multipak upgrade PAL chips, necessary for using the large gray or white Multipaks with the CoCo 3.

< 268'm >

Comments and questions may be sent in care of 68'Micros or directly to :

Dr. Marty Goodman
1633 Bayo Vista Avenue
San Pablo, CA 94806

E-mail: martygoodman@delphi.com

Operating System -Nine

Russell Hoffman

Permanently change drive options, frequently asked questions.

Editor: Well, Rick is a bit busy this time, so we obtained some interesting information to keep readers informed until the next issue.

Floppy Drive Step Rate

The first tid-bit comes from a problem I was having. It seems that no matter how many times I used dmode to change my disk drive step rate, it refused to stay that way! I followed the instructions in the OS-9 manuals, and made sure I used cobbler to make the changes permanent, but they didn't change! It seems that cobbler won't save changes made to the disk drivers using dmode, but there IS a way to easily make the changes permanent:

LOAD MODPATCH

at the prompt, type (don't type comments)-

L/D0 (link to module /d0)

C 14 00 03 (change byte 14 from 00 to 03)

V (verify)

Now use cobbler and the changes will be permanent! Repeat for all disk drives as needed simply by changing the /Dx. 03 changes them to a 6ms step rate.

OS-9 FAQ

This is the sixth edition (23 Apr 1993) of the OS-9 FAQ (Frequently Asked Questions). Addresses will be referenced by number (e.g. [1]) and the address will be listed at the bottom of the file. §§ indicates a question I don't know the answer for. If there are any additions, corrections, suggestions, comments, flames, or contributions, please respond in kind to me, Russell Hoffman, rh2y+@andrew.cmu.edu (editor: I will pass along any comments for those without Internet access).

Q: What is OS-9?

A: OS-9 is a real-time, multiuser, multitasking operating system developed by Microware Systems Corporation. It was originally developed for the 6809 microprocessor, in a joint effort between Microware and Motorola. The original Level I 6809 OS-9 OS was capable of addressing 64 kilobytes of memory. The Level II 6809 OS-9 took advantage of dynamic address translation hardware, and allowed a mapped address space of one megabyte on most systems, and up to two megabytes on others, most notably the Tandy Color Computer III. In the 1980's, Microware ported OS-9 to the 68000 family of microprocessors, creating OS-9/68000. Code is mostly portable from OS-9/6809 to OS-9/68000 at the high-level-language source code level. Code is compatible within either OS-9/6809 or OS-9/68000 at the binary level. OS-9/68000 provides synchronization and mutual exclusion primitives in the form of events, which are similar to semaphores. It also allows communication between processes in the form of named and

unnamed pipes, as well as shared memory in the form of data modules. OS-9 is modular, allowing new devices to be added to the system simply by writing new device drivers, or if a similar device already exists, by simply creating a new device descriptor. All i/o devices can be treated as files, which unifies the i/o system. In addition, the kernel and all user programs are ROMable. Thus, OS-9 can run on any 680x0 based hardware platform from simple diskless embedded control systems to large multiuser minicomputers.

Q: What is OSK?

A: OSK is an abbreviation for OS-9/68000. This is probably due to the common abbreviation '68K' for the 68000 microprocessor. Also sometimes called OS-9/68K.

Q: Where can I get OS-9?

A: Generally the hardware vendor will ship a version of OS-9 with the product upon which OS-9 is intended to be run. Alternatively, OS-9 can be purchased from Microware [1] itself, for certain hardware platforms. In addition, several software vendors sell customized and enhanced OS-9 packages for certain models of computers (see "Nine Flavors of OS-9" in this issue).

Q: What machines run OS-9?

A: OS-9/6809 runs on a variety of platforms, perhaps the most (in)famous being the Tandy Color Computer. Other systems include the SWTPC SCB-69, the Gimix 6809, Smoke Signal Broadcasting's Chieftain 6809, FHL's TC-9, the Febe, and many others, most of which are SS-50 bus machines. Note that OS-9/6809 is no longer supported by Microware, but many user groups, BBSes, and a handful of FTP sites offer help and maintain software collections for OS-9/6809. OS-9/68000 runs on quite a multitude of machines, including a variety of systems from Hazelwood (such as the UniQuad I and II), the Gimix Micro-20, the Atari ST, Commodore Amiga, Apple Macintosh, IMS MM/1, FHL TC-70, and a large number of 680x0-based VME systems, manufactured by such companies as, Motorola, Heurikon, Inducom [12], Gimix [17], Force, Mizar, PEP Modular Computers [14], and others. Gespac [10] also makes machines based on their G-64 and G-96 bus.

Q: Where do I get OS-9/68000 for the Macintosh, Commodore Amiga, and the Atari ST?

A: See "Nine Flavors of OS-9", Vol. 2 No. 3 of "the world of 68' micros"

Q: What is OS-9000?

A: OS-9000 is a portable version of OS-9, written primarily in C. It can potentially run on any 68020 or higher 680x0 family member, and any 80386sx or higher 80x86 member. Code is portable across OS-9000 platforms at the source level. Theoretically, OS-9000 can be ported to any modern computer architecture, though 680x0 and 80x86 are the only supported microprocessor families at present.

Q: What software is available for OS-9?

A: Nearly any user application can be found either commercially or in the public domain/shareware/freeware. Many word-processor, spreadsheets, databases, and time management software packages are available from a variety of vendors. A list of much of the available commercial software is available from Microware. They publish the "OS-9 Sourcebook", a listing of hardware and software vendors who sell both 6809 and 68000 software and hardware. It is advisable to contact the individual companies listed in the Sourcebook and request a recent catalog, as the information in the Sourcebook is a tad outdated. Microware's quarterly magazine Pipelines also carries new product announcements.

Q: Where can I get public-domain/shareware/freeware software for OS-9?

A: There are many private bulletin boards around. In addition, there is the Princeton Listserver, which acts as a mailing server that will mail requested software. To begin using the Listserver, send electronic mail to LISTSERV@ PUCC.PRINCETON.EDU, with the single line HELP in your message. Also, the OS-9 User's Group [7] maintains a library of public domain software, as well as distributes a newsletter. Finally, there are a few anonymous FTP servers worldwide with OS-9 software on them:

cabrales.cs.wisc.edu — Contains mostly OS-9/68000 software, including the complete TOP package, many EFOO disks, GCC and G++, (and many other GNU products such as flex and bison), ka9q, k5jb, TeX, LaTeX, and quite a bit of 6809 software.

wuarchive.wustl.edu — Has mostly 6809 OS-9 software

lucy.ifi.unibas.chMah.rc — Meant to be a European duplicate of cabrales.

Don't forget the often overlooked mailserver on cabrales. The e-mail address is OS-9archive@cabrales.cs.wisc.edu, and a message with "help" as the body will return some help text. This provides some access through mail for those without FTP.

Q: What is the TOP package?

A: TOP is an acronym for "The OS-9 Project". It is a collection of OS-9/68000 software developed primarily in Germany. Much of it seems to be an attempt to make OS-9 a little more UNIX-like. Many standard unix utilities are provided, as well as a complete UUCP mail implementation, and a more secure password file and login program. Many traditional unix games are also provided. The total package consumes approximately 16 MB of disk space, though much of this is source code.

Q: Who/what is the EFFO?

A: EFFO, The European Forum For OS-9, maintains a large library of PD Software for OS-9, e.g. the new C-Kermit 5A(188), an improved KA9Q Software called SLIP for OSK, Ghostscript and many other fine products. The software is available at a nominal handling fee, including written instruction or even a complete users manual.

\$\$ Need EFFO address \$\$

Q: Are there any independent magazines covering OS-9?

A: There is the "OS-9 Underground Magazine" [15], "OS-9 International" [13], and "the world of 68' micros" [16]. OS-9 Underground and 68' micros is in English, OS-9 International is 50% English, 50% German.

Q: Are there alternative shells for OS-9?

A: Yes, there are. Microware sells mshell, an enhanced shell. There are also several public domain shells available. The most notable is the Bourne shell, sh, available in the TOP package (OS-9/68000). It supports aliasing, command-line editing, history, environment variable replacement, shell scripting, the 'command' operator (uses output of command as arguments to called program), and a startup file. A PD version of ksh is available on cabrales. For OS-9/6809, there is Shell+ and of course if you have a Color Computer, there is always Gshell, a graphical shell.

Q: Can one read/write MS-DOS format disks under OS-9?

A: There are several public-domain and commercial utilities to accomplish this task for both OS-9/6809 and OSK. One of the

more interesting is the MSFM file manager which appears in *_OS-9_Insights_*, a book by Peter Dibble, available through Microware. MSFM is an actual file manager, which allows you to mount an MS-DOS floppy as part of the OS-9 file system. For the CoCo, there is a set of utilities called RSDOS and PCDOS that read/writes CoCo (DECB) and MS-DOS disks from OS-9, respectively. They do not format disks though. These are available on the Patch OS-9 disk set from FARNA Systems [16] or on several BBSes.

Q: What sorts of communications software is available?

A: Many public domain utilities, available from your local BBS, include terminal emulators and file transfer utilities (such as xmodem, ymodem, zmodem, and kermit protocols.) Stern, a non-commercial package, also supports Compuserve B+ protocol. Many software vendors sell various equivalent packages. C-kermit is available in source and executable form for OS-9/68000 on cabrales. Microware sells the NFM Network FileManager, which is a local-area networking protocol for small networks of strictly OS-9 based computers. NFM runs on virtually any network interface (direct serial links, ARCnet, Ethernet, and others).

Microware also sells ISP, or Internet Support Package, which is a relatively complete TCP/IP package, including telnet client and server applications, and FTP client and server. It also provides a C BSD 4.2 compatible socket library. Closely related is ESP, or Ethernet Support Package. This is similar to ISP, but is for particular Ethernet boards. Current word from Microware says that the ESP is now obsolete, and has been replaced by a preconfigured version of ISP. ISP supports Ethernet and SLIP, although there is no current SLIP driver supported by Microware. Microware also sells NFS, or Network File System, for OS-9/68000. This allows an OS-9 system to share files in a heterogeneous environment (i.e. not all the machines on the network run OS-9). NFS requires ISP or ESP. Finally, there is a port of the Phil Karn ka9q internet software package, which supports a single-user interface to TCP/IP. It includes a telnet client, an FTP client and server, and SMTP. Source and executables may be found on cabrales. Note that the executables on cabrales have a bug in the FTP server. The k5jb code should correct this.

Q: What about usenet and news?

A: Several ports of UUCP software are available for both OS-9/6809 and OS-9/68K. A port of C news and Rn are available on cabrales. TOP has ported Notes, which

maintains Notesfiles. There is a program which will transfer between Notesfiles and netnews. The TOP package in its entirety may be found on cabrales. Rick Adams' UUCP port for the Color Computer may be found on wuarchive, as well as on Delphi and Compuserve.

\$\$ Need info on the latest ELM port \$\$

Q: Is gcc available for OS-9?

A: gcc and g++ are available for OS-9/68000, both in OS-9 executable form and cross-compiler form. Versions 1.37, 1.39 and 1.40 were ported to OS-9/68000 primarily via the work of Stephan Paschedag. Source and binaries are available on cabrales.cs.wisc.edu via anonymous FTP. The 1.40 version supports 68040 optimizations.

Q: Can I run X11 on OS-9?

A: Yes. Microware [1] sells a port of X11R4 (client and server plus optional Motif), as well as do Eltek Electronik GmbH [8]

\$\$ Need info on Keith Thompson's X11R5 port \$\$

Q: What other graphics alternatives are there?

A: Several other organizations have various graphics packages for OS-9. ReccoWare Systems [9] has a port of the Bellcore MGR window manager. Gespac [10] produces G-windows, a portable windowing package which has device windows and a very Motif-looking interface. For the MM/1, BlackHawk Inc. [11] is producing K-windows, a window manager similar to Multi-Vue, the OS-9 window package for the Tandy Color Computer III. Microware [1] also sells RAVE, the Real-Time Audio Video Environment.

Q: What is a Real Time system?

A: A real-time system is any system whose correctness depends not only on the correctness of the applied algorithms, but also in the timing of the execution of those algorithms. Refer to the *netnews comp.realtime newsgroup* for more information.

Q: Are there analytical frameworks for reasoning about timing correctness of real-time applications written under OS-9, ala Lohcsky, Sha, and Ding analysis?

A: PLEASE! Somebody find out! Otherwise, I'll be forced to make an OS-9 operating system model myself...

Q: Does OS-9 support multiple threads within a program?

A: Not directly, but through the use of user installed periodic interrupts or alarms, a program can support its own threads.

FAQ Addresses, by reference number:

1
Microware Systems Corporation
1900 N.W. 114th Street
Des Moines, Iowa 50322
Phone: (515) 224-1929
Fax: +1 (515) 224-1352

Microware Systems Corporation
Western Regional Office
2041 Mission College Boulevard
Santa Clara, California 95054
Phone: +1 (408) 980-0201
Fax: +1 (408) 980-1671

Northeastern Regional Office
One Crank Rd
Hampton Falls, NH 03844
(603)929-4107
(603)929-4233 fax

Southeastern Regional Office
P.O. Box 510358
Melbourne Beach, FL 32951-0358
(407)725-2840
(407)725-2487 fax

Microware Systems (UK) Limited
Leylands Farm, Nobs Crook
Colden Common
Winchester, Hants.
England, SO21 1TH
Phone: +44 703 601990
Fax: +44 703 601991

Microware Systems K.K.
17-3, Sotokanda 2-Chome
Chiyoda-Ku
Tokyo 101, Japan
Phone: +81 3-3257-9000
Fax: +81 3-3257-9200

Microware Systems France
Chateau de la Saurine
Pont de Bayeux
13590 Meyreuil
France
Phone: +33 42 58 63 00
Fax: +33 42 58 62 28

2
OS-9 User's Group, Inc.
6158 W. 63d St., Suite 109
Chicago, IL 60638

3
Eltek Electronik GmbH
Galileo-Galilei-Strasse 11
D-6500 Mainz 42
Germany
Phone: (6131) 588-0
fax: (6131) 588-199

4
Reccoware Systems
Wolfgang Ocker
Lochhauser Strasse 35a
D-8039 Puchheim
Voice: +49 89 80 77 02
Fax: +49 89 80 29 67

5
To contact Gespac, call toll-free
1-800-4GESPAC

6
BlackHawk Enterprises, Inc.
P.O. Box 10552
Enid, OK 73706-0552
Voice: 405-234-2347
E-mail: nimitz@delphi.com

7
Inducom Systems BV
PO Box 627
NL 5340 AP OSS
Phone (31)4120-41922
Fax (31)4120-22640

8
OS-9 International
Hagentalerstrasse 12
CH-4055 Basel, Switzerland
Tel. +41 61 43 55 01
Fax +41 61 43 55 02
E-mail OS-9int@msys.ch

9
PEP Modular Computers
USA
Pep Modular Computers, Inc.
750 Holiday Drive, Building 9
Pittsburgh, PA 15220
Tel ++1 (0) 412 921-3322
Fax ++1 (0) 412 921-3356
Toll Free 800-228-1737

10
Great Britain
PEP Modular Computers Ltd.
Unit 3, Riverside Business Centre
Brighton Road
Shoreham-By-Sea
West Sussex, BN43 6RE
Tel ++44 (0) 274 44 11 88
Fax ++44 (0) 273 44 11 99

11
Benelux
PEP Modular Computers S.A.
Koning Albert 1-straat 46
B-1780 WEMMEL
Tel ++32 (0) 24 61 04 08
Fax ++32 (0) 24 61 00 31

12
Germany
PEP Modular Computers GmbH
Apfelstrasse 16
D-8950 KAUFBEUREN
Tel ++49 (0) 83 41 43 02 - 0
Fax ++49 (0) 83 41 43 02 - 39

13
OS-9 Underground
4650 Cahuenga Blvd. Ste 7
Toluca Lake, CA 91602
Voice: 818-761-4135

14
the world of 68' micros
FARNA Systems
Box 321
Warner Robins, GA 31099-0321
Voice: 912-328-7859
E-mail: dsrtfox@delphi.com

15
GMX (Gimix)
3223 Arnold Lane
Northbrook, IL 60062
Voice: 708-559-0909

< 268'm >

Comments and questions may be sent in care of 68'micros or directly to the author at:

Rick Ulland
449 South 90th
West Allis, WI 53214
E-mail is rickuland@delphi.com

Comments, additions, etc., for the OS-9 FAQ should be sent to:

rh2y+@andrew.cmu.edu
(Russell Hoffman)

Those without Internet access may send FAQ comments to 68'micros. We will forward the messages for you.

OS-9/OSK Answers!

Tackling TERMCAP

One of the most useful concepts used under OS-9 is "termcap", which is short for terminal capabilities. Within this standard, all the terminal-specific control sequences and special-key strings are kept within a dd/SYS/termcap file. This allows your software to run on any terminal, whether it's VT-100, ANSI, K-Windows, ADM3A, Hazeltine, ABM85, TVI920... well, you get the idea. There are so many "standards" for terminals that it is obviously a tremendous benefit to be able to obtain this information easily at run-time for the type of terminal the user is on.

As it turns out, this concept is also one of the most difficult to implement! Those of you who have tried to use termcap know what I mean. For those who've only glanced at it, this may come as a surprise. After all, the OS-9 C compiler comes with an entire library (termlib.l) dedicated to helping C programmers implement termcap within their software. The problem is the library doesn't handle the tough stuff. Many complexities are left for the programmer to handle, and the documentation for the termcap library is scarce, severely lacking in detail, and even self-contradictory in some parts. What I hope to do in the next couple issues is fully explain termcap to the fullest and give everyone the tools and knowledge to use termcap in their software.

Due to space considerations, I'm only going to present the source-code for an example program in this issue. I could have done it the other way around (explained what I want to do and then give the source in the next issue), but I know how programmers prefer to get experimenting with actual source code! The program is fairly large for what it does, which shows just how complex it is to implement in even just a simple manner.

The program, 'termcaptest.c' (listing #2), merely clears the screen for the user and lets him/her type anywhere on the screen. The arrow keys may be used to move the cursor around the screen, and the backspace string is also supported.

When you want to quit the program, press your keyboard interrupt key (usually control C or control E) and the program will clear the screen and exit back to OS-9.

This program requires the following terminal capabilities to be defined with in the termcap file: up, cl, cm, ku, kd, kl, kr, kb. If any of these are missing, you will receive an "incomplete termcap entry" message. For more information on editing your termcap file and what to put inside it, read pages 8-31 through 8-37 in your "Using Professional

OS-9" manual. I've included my termcap entry for my K-Windows system (vsc) as listing #1.

Next issue, I'll fully explain how the program works and give some insight on what the library functions within termlib.l actually do. You will then be well on your way to understanding termcap and employing this powerful standard with in your own software!

Listing #1: example termcap entry

```
k1\vsclSignetics Vsc Video driver by RMC\
:am:bs:cl=^L:li#26:co#80:ho=^A:\
:cd=^K:ce=^D:cm=^B%r%+ %+ :pt\
:do=^J:up=^E[A:nd=^F:so=037
:se=037!\
:us=037":ue=037#:al=0370:dl=0371\
:ku=^P:kd=^N:kr=^F:kl=^B:kb=^H:sf=012:\
:ve=005!:vi=005 :bl=007:
```

Listing #2: termcaptest.c

```
#include <stdio.h>
#include <sgstat.h>
#include <termcap.h>

#define TCAPSLEN 400
#define BUFFSIZE 20
#define STDIN 0
#define STDOUT 1
/* user-defined values for special keys */
/* values are not important, but must */
/* be greater than 255. ENTER is just */
/* the value 13 for CR. */
#define UP_ARROW 400
#define DOWN_ARROW 401
#define LEFT_ARROW 402
#define RIGHT_ARROW 403
#define BACKSPACE 404
#define ENTER 0x0D
extern char *getenv();
int user_tputc(), user_touts(), putpad(), sigtrap();

/* variables for termcap stuff */
char tcapbuf[TCAPSLEN];
short lines,columns,ospeed;
char PC_,
    *UP, /* required by tgoto() */
    *BC, /* required by tgoto() */
    *CL,
    *CM,
    *KU,
    *KD,
    *KL,
    *KR,
    *KB;

/* variables for my program */
int keybuslen,keybuspos,keyflag=0, done=0;
char keybuff[BUFFSIZE];
struct sgbuf sgbuffer,oldbuffer;
main()
{
    int keypress,
        cx=0,cy=0; /* initial cursor position */
    char keychar;

    /* program initialization */
    initialize_termcap();
    intercept(sigtrap);
    getstat(0,STDIN,&sgbuffer);
    _strass(&oldbuffer,&sgbuffer,32);
    sgbuffer.sg_pause=0;
    sgbuffer.sg_echo=0;
    sgbuffer.sg_eofch='0';
    setstat(0,STDIN,&sgbuffer);

    /* clear screen & position cursor */
    putpad(CL);
    putpad(tgoto(CM,cx,cy));

    while(!done)
    {
        if (keyflag==1) /* old keys waiting */
            keypress=process_key();
        else
        {
            /* get a key & process it */
            read(STDIN,&keychar,1);
            keypress=process_key(keychar);
        }

        if (keypress)
        {
            switch(keypress)
            {
                case UP_ARROW:
                    if (--cy<0) cy=0;
                    break;
                case DOWN_ARROW:
                    if (++cy>=lines) cy=lines-1;
                    break;
                case LEFT_ARROW:
                    if (--cx<0) cx=0;
                    break;
                case RIGHT_ARROW:
                    if (++cx>=columns) cx=columns-1;
                    break;
                case BACKSPACE:
                    if (--cx<0) cx=0;
                    putpad(tgoto(CM,cx,cy));
                    tputc(' ');
                    break;
                case ENTER:
                    cx=0;
                    if (++cy>=lines) cy=lines-1;
                    break;
                default:
                    /* make sure character is printable*/
                    if (keypress>=0x20 &&
                        keypress<=0x7e)
                    {
                        /* put it on the screen */
                        tputc((char)keypress);
                    }
            }
        }
    }
}
```

Joel Mathew Hegberg



```

    if (++cx>=columns)
    {
        cx=0;
        if (++cy>=lines) cy=lines-1;
    }
    break;
}
/* make sure cursor is where it should be*/
putpad(tgoto(CM,cx,cy));
}
/* cleanup before exit, clear screen */
putpad(CL);
/* home cursor position */
putpad(tgoto(CM,0,0));
/* reset stdin characteristics */
setstat(0,STDIN,&oldbuffer);
}

/* process_key is given character to be */
/* processed. The processed character */
/* (of type int) is returned or zero (0) */
/* if key was swallowed. */
/*keyflag may have the following values:*/
/* -1 = old keys removed from buffer. */
/*  0 = no keys are in buffer. */
/*  1 = keys being swallowed into buffer.*/
int process_key(key)
char key;
{
    int retkey,t;
    switch(keyflag)
    {
        case -1:
            /* remove next key from buffer */
            retkey=(int)keybuff[keybufpos++];
            if (!(--keybuflen)) keyflag=0;
            break;
        case 0:
            keybuff[0]=key;
            keybufpos=1;
            keybuflen=1;
            t=compare_special();
            if (!t) retkey=(int)key;
            else
            {
                if (t==1)
                {
                    retkey=0;
                    keyflag=1;
                }
                else retkey=t;
            }
            break;
        case 1:
            keybuff[keybufpos++]=key;
            keybuflen++;
            t=compare_special();
            if (!t)
            {
                keyflag=-1;
                retkey=keybufpos=0;
            }
            else
            {
                if (t==1) retkey=0;
                else
                {
                    if (++cx>=columns)
                    {
                        cx=0;
                        if (++cy>=lines) cy=lines-1;
                    }
                    break;
                }
            }
            return(retkey);
    }
}

/* compare_special checks to see if the */
/* string within the buffer matches, partially*/
/* matches, or does not match any of the */
/* special-keys supported by termcap. The */
/* return value will be: 0= no match at all. */
/* 1= partial match, swallow more chars.*/
/* ? = user-defined value of special key.*/
int compare_special()
{
    int t=0;
    keybuff[keybufpos]='\0';
    /* check for exact match */
    if (!strcmp(keybuff,KU)) t=UP_ARROW;
    if (!strcmp(keybuff,KD)) t=DOWN_ARROW;
    if (!strcmp(keybuff,KL)) t=LEFT_ARROW;
    if (!strcmp(keybuff,KR)) t=RIGHT_ARROW;
    if (!strcmp(keybuff,KB)) t=BACKSPACE;
    if (t) return(t);

    /* check for partial match */
    if (!strncmp(keybuff,KU,keybuflen)) t=1;
    if (!strncmp(keybuff,KD,keybuflen)) t=1;
    if (!strncmp(keybuff,KL,keybuflen)) t=1;
    if (!strncmp(keybuff,KR,keybuflen)) t=1;
    if (!strncmp(keybuff,KB,keybuflen)) t=1;
    return(t);
}

initialize_termcap()
{
    char tcbuf[1024],*term_type,*temp,*ptr;
    if ((term_type=getenv("TERM"))==NULL)
    {
        sprintf(stderr,"Environment variable TERM\nnot defined.\n");
        exit(1);
    }
    if (tgetent(tcbuf,term_type)<=0)
    {
        sprintf(stderr,"Unknown terminal type\n'%s'.\n",term_type);
        exit(1);
    }
    /* read the termcap entry */
    ptr=tcapbuf;
    if (temp=tgetstr("PC",&ptr)) PC_=*temp;
    CL=tgetstr("cl",&ptr);
    CM=tgetstr("cm",&ptr);
    KU=tgetstr("ku",&ptr);
    KD=tgetstr("kd",&ptr);
    KL=tgetstr("kl",&ptr);
    KR=tgetstr("kr",&ptr);
    KB=tgetstr("kb",&ptr);
    UP=tgetstr("up",&ptr);
    lines=tgetnum("li");
    columns=tgetnum("co");
    ospeed=-1; /* no padding */

    if (lines<1 || columns<1)
    {
        fprintf(stderr,"No rows or columns!\n");
        exit(1);
    }

    /* make sure we have everything */
    if (!(CL && CM && KU && KD && CL
    && KR && KB && UP))
    {
        fprintf(stderr,"Incomplete termcap\nentry.\n");
        exit(1);
    }

    if (ptr>=&tcapbuf[TCAPSLLEN])
    {
        fprintf(stderr,"Terminal description too\nbig!\n");
        exit(1);
    }

    /* writes one character to terminal. */
    /* needed by tputs() library function. */
    int user_tputc(c)
    char c;
    {
        return (write(STDOOUT,&c,1));
    }

    /* writes an entire string to terminal. */
    user_touts(s)
    char *s;
    {
        write(STDOOUT,s,strlen(s));
    }

    /* writes out a special sequence */
    /* to the terminal. */
    putpad(s)
    char *s;
    {
        tputs(s,1,user_tputc);
    }

    /* when a signal is received, we're done! */
    sigtrap(signal)
    int signal;
    {
        done=1;
    }

    /* THE END */ < 268'm >
}

Any comments, questions, or source
code to be included in Joel's column
may be sent in care of 68'Micros or
directly to Joel at:
Joel Mathew Hegberg
936 N. 12th Street
DeKalb, IL 60115
E-mail : joelhegberg@delphi.com

```


WILDCARD.BAS

Charles R. Connolly

Wildcard routine for last issue's SDISK512K program

```

10 ' WILD.BAS
20 'Wildcard file marking program called
30 ' by Disk Organizer "SDISK512K.BAS".
41 ' (c) Charles R. Connolly 1994
50 CLS:PCLEAR 1
60 PALETTE CMP:ATTR 0,0
70 CLEAR 3000,30000
80 FILES 3,768
90 DIM F$(68),E$(68),T$(68),A$(68),CE$(10)
100 ' F$=File Name E$=File Extension T$=
File Type A$=Ascii Flag.
110 DIM SG(68),NG(68),FSL(68),EC(68),OC
(68),GC(7)
120 ' SG=Start Granule NG=Number of Gran
ules GC=Granules to Copy
130 DIM HH$(64),HO(8),WS(8),AT(8)
140 'HELP VAR'S HH$=Text HO=Offset WS
=Window Size AT=Attribute
150 QM=2:CO=0:'Query Mode,Scratch Disk,
Program Disk & CO=Context
160 SE=3
170 POKE 30100,78;"N" DO NOT REDO
THE FILE MANAGER STARTUP SCREEN
180 PD=PEEK(30101)
190 NF=0:Number of Files.
200 MD=3:Max Drive Number.
210 IF PD=0 THEN OPEN "I",#1,"MENU-
EXT.DAT:0"
220 IF PD=1 THEN OPEN "I",#1,"MENU-
EXT.DAT:1"
230 IF PD=2 THEN OPEN "I",#1,"MENU-
EXT.DAT:2"
240 IF PD=3 THEN OPEN "I",#1,"MENU-
EXT.DAT:3"
250 EN=1
260 'EXTENSION FILE LOOP
270 :IF EN>9 THEN GOTO 320
280 :IF EOF(1)=-1 THEN GOTO 320 ELSE
INPUT #1,YE$
290 :SC$=MID$(YE$,1,1)
300 :IF SC$="" THEN CE$(EN)=MID$(YE$,
2,3):EN=EN+1:GOTO 260
310 :IF SC$="" THEN GOTO 260
320 'WEND
330 ME=EN
340 CLOSE #1
350 PRINT
360 PRINT "INSERT DATA DISK IN ":"D
370 PRINT "PRESS ANY KEY"
380 AK$=INKEY$:IF AK$="" THEN 380
390 WIDTH 32
400 PRINT:PRINT:PRINT "reading directory
":PRINT
410 'REPEAT
420 :DSK$ D,17,SE,SL$,SH$
430 :SEC$=SL$+LEFT$(SH$,127)
440 :FORX=0TO7
450 ::Z=X*32: Each Directory Entry is 32 Ch
aracters Long.
460 ::IF MID$(SEC$,Z+1,1) = CHR$(255) GO
TO 510
470 ::IF MID$(SEC$,Z+1,1)<>CHR$(0) THE
N GOSUB 1060
480 :NEXT X
490 :SE=SE+1
500 IF SE<12 GOTO 410:'UNTIL

510 NF=NF-1:F$(NF+1)=CHR$(255)
520 FOR I= 0 TO 68:OC(I)=32:NEXT
530 FOR I=0 TO 68:EC(I)=0:NEXT
540 ' FIND THE SIZE OF THE FILE IN GRA
NULES
550 DSK$ D,17,2,SL$,SH$
560 FOR X=0 TO NF
570 :G=1:GL=SG(X)+1
580 :IF MID$(SL$,GL,1)>HEX$(191) THEN
NG(X)=G:GOTO 630
590 ::G=G+1:While NOT last Granule In Th
e File
600 :GL=ASC(MID$(SL$,GL,1))+1
610 ::IF MID$(SL$,GL,1)>HEX$(191)THEN
NG(X)=G:GOTO 630
620 :GOTO 590
630 NEXT
640 ' WILD initialization
650 OF=PEEK(30199):See if "Y"es.
660 IF OF<>89 THEN PRINT "ARRAY NO
T READY":END
670 AD=PEEK(30198)
680 IF AD<>D THEN PRINT "INCORECT
DRIVE":END
690 FOR I=0 TO NF:OC(I)=PEEK(30200+I):
NEXT
700 'WILD main program loop.
710 CLS:PRINT " WILD CARD BUILDER"
:PRINT
720 PRINT " code  operation"
730 FOR I=0 TO MD
740 :IF I<>D THEN PRINT USING " ##
copy FROM DRIVE ## TO ##";I;D;I
750 NEXT
760 PRINT " K kill"
770 PRINT " R rename"
780 PRINT " U undo"
790 PRINT:INPUT " OPPERATION CODE"
:OP$
800 IF OP$="U" THEN OP$=" UNDO"
810 -- REM ADD ERROR CHECKING !!
820 CLS:PRINT " OP = ";OP$
830 PRINT:PRINT "MARK aLL OR sELEC
T (a/S) ";
840 INPUT MT$
850 IF MT$="A" OR MT$="a" THEN GOS
UB 1980:GOTO 1000:'MARK ALL
860 'SELECTIVE MARK.
870 GOSUB 2150
880 PRINT " SELECT BY:"
890 PRINT:PRINT " FILE nAME"
900 PRINT " eXTENSION"
910 PRINT " FILE tYPE"
920 PRINT " QUIT"
930 PRINT
940 INPUT " CRITERIA ";CS$
950 GOSUB 2150
960 IF CS$="N" OR CS$="n" THEN GOS
UB 1170
970 IF CS$="E" OR CS$="e" THEN GOSU
B 1320
980 IF CS$="T" OR CS$="t" THEN GOSU
B 1450
990 IF CS$="Q" OR CS$="q" THEN
GOTO 2030
1000 'DONE ?
```

```

1010 CLS:PRINT " MORE (y/N) "
1020 INPUT MW$
1030 IF MW$="N" THEN GOTO 2030
1040 GOTO 700
1050 END
1060 'SUB Get a Disk Directory Entry's Cont
ents.
1070 ::F$(NF)=MID$(SEC$,Z+1,8): The File
Name is the first 8 Bytes of the Entry.
1080 ::E$(NF)=MID$(SEC$,Z+9,3): The File
Extension is the next 3 Bytes.
1090 ::IF MID$(SEC$,Z+12,1)=CHR$(0) THE
N T$(NF)="0" ELSE IF MID$(SEC$,Z+12,1)
= CHR$(1) THEN T$(NF)="1" ELSE IF MID$(
SEC$,Z+12,1) = CHR$(2) THEN T$(NF)="2"
ELSE T$(NF)="3"
1100 ::IF MID$(SEC$,Z+13,1)=CHR$(0) THE
N A$(NF)="B" ELSE A$(NF)="A"
1110 PRINT
1120 ::SG(NF)=ASC(MID$(SEC$,Z+14, 1))
1130 ::NF=NF+1
1140 RETURN
1150 END
1160 CLS:GOTO 700
1170 'MARK FILES BY NAME
1180 PRINT " MARK FILES BY NAME": PR
INT
1190 PRINT " ? n n* *n":PRINT
1200 INPUT " ";WN$
1210 IF WNS="?" THEN GOSUB 1670: 'HE
LP
1220 WF$=LEFT$(WN$,1):WE$= RIGHT$(
WN$,1)
1230 IF WF$="?" THEN GOSUB 1670: GO
SUB 2150:GOTO 1180
1240 IF WF$="" THEN GOSUB 1790: RET
URN
1250 IF WE$="" THEN GOSUB 1890: RET
URN
1260 'EXACT MATCH
1270 FOR SI=0 TO NF
1280 :FS$=F$(SI)
1290 :IF WNS=FS$ THEN OC(SI)=ASC(OP$)
1300 NEXT
1310 RETURN
1320 'MARK FILES BY EXTENSION
1330 CLS
1340 FOR EI=1 TO ME:PRINT EI;"+CE$(
EI):NEXT
1350 PRINT 0;"NONE OF THE ABOVE" :P
RINT
1360 INPUT WN$
1370 CI=ASC(WN$)-48
1380 IF WN$="0" THEN INPUT " EXTENS
ION ";EX$ ELSE EX$=MID$(CE$(CI),1,3)
1390 PRINT " ..:MATCHING EXTENSION"
:EX$
1400 FOR SI=0 TO NF
1410 :ES$=E$(SI)
1420 :IF EX$=ES$ THEN OC(SI)=ASC (OP$)
1430 NEXT
1440 RETURN
1450 'MARK FILES BY TYPE
1460 PRINT "ASCII OR BINARY OR n/A :"
1470 WN$=INKEY$:IF WN$="" THEN GOT
O 1470
1480 IF WN$="A" OR WN$="B" OR WN$=
="a" OR WN$="b" THEN GOTO 1560
1490 'TYPE ONLY

```

continued on page 25

Programming in "C"

The C preprocessor

P.J. Ponzo

```
BEGIN          main() {  
DECLARE_AN_INTEGER x;    int x;  
LET x=0;          x=0;  
AS_LONG_AS ( x IS_LESS_THAN 5 ) DO  while (x<5) {  
LET x INCREASE_BY 1;      x+=1;  
PRINT_x;                  printf("%d",x);  
THEN_STOP                }  
END
```

Wouldn't it be nice if we could write the Left-Hand program (above) and have it (magically) turn itself into the Right-Hand program?

This would require the following definitions:

```
#define BEGIN  to mean main() {  
define DECLARE_AN_INTEGER to mean int  
define LET  to mean (nothing!)  
define AS_LONG_AS  to mean while  
define IS_LESS_THAN  to mean <  
define DO  to mean {  
define INCREASE_BY to mean +=  
define PRINT_x  to mean printf("%d",x)  
define THEN_STOP  to mean }  
define END  to mean }
```

One fascinating aspect of the C language is its ability to perform replacements of BEGIN by main() {, and LET by nothing (a "null string") and END by }, etc. etc.

Before the C-compiler goes to work on your program, a C-preprocessor will make the appropriate replacements ... and (MAGIC!) the compiler gets a normal, standard, C program to compile. To tell the preprocessor that you have redefined certain C-phrases, you need only begin your program with these #definitions.

#define THIS and THAT

```
#define BEGIN  main() {  
define DECLARE_AN_INTEGER int  
define LET  
define AS_LONG_AS  while  
define IS_LESS_THAN  <  
define INCREASE_BY  +=  
define PRINT_x  printf("%d",x)  
define THEN_STOP  }  
define END  }
```

```
BEGIN  
DECLARE_AN_INTEGER x;  
LET x=0;  
AS_LONG_AS ( x IS_LESS_THAN 5 ) DO  
PRINT_x;  
THEN_STOP  
END
```

If the above program is written and compiled, the preprocessor will make all replacements indicated by the various #define statements!

The #define statements are instructions to the preprocessor. We've actually seen such instructions before! Remember #include <stdio.h> ?? THAT was also an instruction to the preprocessor .. to replace the phrase, #include <stdio.h> by the library of standard input/output routines (.. which explains why a compiled program is often very much larger than the program we write with our favourite text-editor .. it contains the i/o library of functions!).

#include <stdio.h> and #include <math.h>

Since C is supposed to be a PORTABLE language (write a C program on your favourite text-editor and compile it for ANY computer for which you have a compiler), the standard input/output routines are system dependent ... and not, strictly speaking, part of the language! Somebody has written the functions in stdio.h with a specific computer in mind... which reminds me ... the library of MATH functions (such as sin()) is NOT in stdio.h. If you use them, be sure to #include <math.h>.

Even Sexier #Definitions

In the previous example we used:

```
#define PRINT_x  printf("%d",x)
```

Which meant that, to print y or print a etc. we would need more #definitions (like PRINT_y and PRINT_a). BUT, C is very smart. You can include a VARIABLE in the #definition (within parentheses!), for example: #define PRINT(x) printf("%d",x)

... and now use PRINT(y) or PRINT(a) etc and the preprocessor would make the substitutions to.

QUESTION: How to teach programming to kids?

ANSWER: Invent your own kiddie-C!

Let the kids write this: but have the C-compiler get this:

```
BEGIN          main() {  
LET x=0;          int x=0;  
IF (x LESS 10) DO  while (x<10) {  
    PRINT(x);          printf(" %d ",x);  
    PRINT(SQUARE(x));  printf(" %d ",x*x);  
    NEWLINE;          printf("\n");  
    INCREASE(x);      x+=1;  
  ENDIF  
STOP          }
```

... can you see what #definitions are necessary ??? These #definitions:

```
#define BEGIN  main() {  
define LET  int  
define IF  while  
define LESS  <  
define DO  {  
define PRINT(x)  printf(" %d ",x)  
define SQUARE(x) x*x  
define NEWLINE  printf("\n")  
define INCREASE(x) x+=1  
define ENDIF  }  
define STOP  }
```

turn this:

```
BEGIN  
LET x=0;  
IF (x LESS 10) DO  
  PRINT(x);  
  PRINT(SQUARE(x));  
  NEWLINE;  
  INCREASE(x);  
ENDIF  
STOP
```

into this:

```
main() {  
int x=0;  
while (x<10) {  
  printf(" %d ",x);  
  printf(" %d ",x*x);  
  printf("\n");  
  x+=1;  
}
```

Of course the kids would have to type in all those !@#\$% #definitions... or would they ??

ROMY-16

EPROM EMULATOR

- Emulates ROMS (2716-27010) or RAMs
- in 8- and 16-bit systems.
- Window/menu driven interface.
- Provides 8 hardware breakpoints for 8-bit systems.
- **\$195** (2716-27256) or **\$245** (2716-27010), 90 day warranty.
- 15 day money-back guarantee.
- Optional assembler, disassembler, and

ROM debugger add \$100

Universal Microprocessor Simulator/Debugger V2.1

- Simulates Z8, Z80, 64180, 8048, 8051 8085, 6800, 6801, 6805, 6809, 68HC11, 6303, 6502 & 65C02.
- Assembler, Disassembler, & Windowed Symbolic Simulator. Supports on-board debug through RS232.
- **\$100** each CPU (S&H \$8)

6809 Single Board Computer

- Supports 8K RAM, 32K ROM.
- Two 6821 PIAs connect 32 bits of I/O to outside world.
- No jumpers for 2732, 2764, and 6116.
- Two interrupt signals on CPU bus.
- Size is 2.75" x 5". **\$60** each board.
- For an integrated development system with assembler, disassembler, and on-board debugger please add \$70.

68HC11 Microcontroller Development System

- Eight channel 8-bit A/D converter.
- 32K ROM and 32K RAM.
- \$120 each SBC, to complete with assembler, disassembler, BASIC interpreter and on-board debugger add \$70

J&M Microtek, Inc.

83 Saman Road, W Orange, NJ 07052
Tel: 201-325-1892 Fax: 201-736-4567

- CoCo-C is still available, but we only have about a dozen left!
- Send **\$25.00** (plus \$2.50 S&H) for your copy today, and start learning to program in C on your CoCo (all models supported) without OS-9!

FARNA Systems

Box 321

Warner Robins, GA 31099

We (the master over 'C'er) would have all those !@#\$% #definitions in a separate file (on disk) called sugar.c so the kids would only have to begin their program with: #include <sugar.c> The preprocessor would #include all those !@#\$% #definitions at the beginning of every program, then make all the appropriate substitutions/replacements ...

We have written the #definitions using CAPITALS to replace C-phrases. That's not necessary, but ADVISABLE (then it's obvious which parts of your program are HOME-MADE and which are "real C")! Not only do #definitions allow us to write C in a personal dialect, but they also allow us to quickly change certain C-phrases.

Notes from the author:

On the IBM PC (with the "ansi.sys" program installed), I can clear the screen by using : printf("?[2J")

In place of the question mark, I type an "escape character" by holding down the Alt key and typing the number 27 on the number pad ... and the "left-arrow" appears when I release the Alt key. Now, I begin my programs with a #definition (among others)

#define CLEAR printf("?[2J")
then I can use CLEAR; wherever I need it, in my program.

Tomorrow, when I buy a sexier C-compiler for the PC (or learn more C!)... one which has all the features included in "ansi.sys" ... including a clr_screen() function, I replace the above #definition by:

#define CLEAR clr_screen()

TypeDef(initions)

You begin your program with #define SAM int and it will be preprocessed, and every occurrence of SAM will be replaced by int... as in SAM x=0; which getss passed on to the compiler (by the preprocessor) as int x=0;

This "replacement" by the preprocessor holds for anything ... not just DATA TYPES (so #define NewLinePlease printf("\n") allows you to use NewLinePlease; in place of printf("\n");).

BUT, there is a "special" feature for DATA TYPES, called **typedef**. Place, at the beginning of your program:

typedef int INCHES (note the order!)

... and you may subsequently use, in your program:

INCHES x, y, z;

and the variables x, y and z will be regarded (by the compiler) as int variables.

typedef int INCHES (note the order!)

... doesn't seem like it's any different from:

#define INCHES int (note the order!)

BUT, remember the curious way we had to refer to an argument, passed to a function, which was itself a function?

```
float solve(f,x,error)
float (*f)();           (note the curiosity!)
float x, error;
{
```

Here, the C-phrase float (*f)(); says that f is a pointer to a function, and this function returns a float.

```
float solve(f,x,error)
float (*f)();           (note the curiosity!)
float x, error;
{
```

Now, using **typedef**, we may introduce a home-made DATA TYPE called Pointer to a Function which returns a Float.

typedef float (*PFF)() (note the order!)

Use the above **typedef** (near the beginning of your program) and (MAGIC!) the phrase PFF is interpreted (by the compiler) as meaning a Pointer to a Function which returns a Float. Use it, in solve(), like so:

```
float solve(f,x,error)
PFF f;
float x, error;
{
```

Since **typedef** is more than "just a dumb replacement of characters", the preprocessor won't touch it ... but passes it along to the compiler! And THAT makes it different from a #define too!

TypeDef Char *STRING

Here STRING is a pointer to a string and you may write STRING x; meaning that x is a pointer to a character string.

typedef int VECTOR[5]

Here VECTOR is an array of 5 ints and you may write VECTOR x; so x is an array of 5 integers.

typedef int INTEGER

Here INTEGER is an int (of course!), and you may write INTEGER x; ... and, when you compile your C program for a different computer where int DATA TYPES are 4 bytes long (and you were counting on their being only 2 bytes long!) you change this one statement to:

typedef short INTEGER

meaning a "short" integer.
and all your references to INTEGER x; and INTEGER y; etc. will be changed!

< 268'm >

P.J.Ponzo
Dept. of Applied Math
Univ. of Waterloo
Ontario N2L 3G1
CANADA

Basic09 In Easy Steps

Chris Dekker

Running machine language subroutines from within Basic09.

Once your ML subroutine is written and debugged, running it can be accomplished with a simple RUN statement in the same way as you run Basic09 subroutine modules. Before we get to that point, however, some work has to be done and that's today's focus.

Why bother with ML subroutines?? There are two basic reasons: Either you can't get the job done in Basic09 or you can't get it done fast enough. When presented with one of those problems you don't have to go running for your assembler right away, though. There are a number of ways that may solve your problem quicker and easier.

For starters you may be able to rewrite parts of your Basic09 program to speed things up (something I will deal with in a separate article). Your next best bet is to search the technical reference section of the OS-9 manual for a documented system call that will do the job. A lot of times this will be the case and you will only have to provide some pointers and/or codes to get the job done through the SYSCALL utility.

Mind you, outside these avenues there are plenty of occasions left to write an ML subroutine. A few that come to mind are: no system call will do the job, using a system call is too cumbersome, you need an improved version of a Basic09 function or you simply have to do a job that OS-9 wasn't designed to do.

To get a little more specific: Let's say you need to move a few thousand bytes. If they have to be moved from one string to another or from one array to another you can do a quick and easy job with Basic09 itself. Just use the = operator as in scratch=buffer. Assuming "buffer" and "scratch" are identical in size and type this will work fine. But what if one is an array of bytes, while the other is defined as a string? In that case Basic09 will generate an error and refuse to let you use this setup.

So we could use the F\$move system call. However since this call is designed to access the computer's entire memory it wants to see a DAT image so it can assemble an extended address. Although not impossible to get, this makes the entire process a little too cumbersome for my taste. At least when it comes to moving some bytes inside Basic09's 64K address space.

So what about simply PEEKing and POK(E)ing the bytes with Basic09. This works fine for a small array but anything over, say, 500 bytes will noticeably slow down your CoCo every time it has to copy the data. In real numbers: you will be lucky to transfer about 5000 characters per second.

By comparison a ML routine, even with the overhead of starting a separate module, will reach transfer speeds of 30,000 to 40,000 characters per second depending on exactly how it is coded. Of course a 6309 is still an order of magnitude faster. After setting up it's registers, the chip's internal hardware takes over, transferring data at a clip of 1 character (or byte) per 3 clock cycles. This amounts to almost 600,000 characters per second.

To give you some examples of other reasons for using ML subroutines: decoding graphics data into printer data is awfully slow in Basic09 because of the huge amounts of data that can be involved. For instance, to make a full color printout of a 320x192 pixel 16 color screen in 24-bit high density mode on my Star NX2420 printer, the computer generates a data stream of approximately 310K bytes. Also prime candidates are data compression/expansion, searching buffers on a byte by byte basis, stripping byte sequences out of textfiles, etc.

Another good reason is to improve on existing Basic09 functions. For instance SUBSTR does a great job to search strings, but it is case sensitive. So what do you do when you need a function that is not case sensitive? You write one.

The last example I want to mention is that of a Basic09 program that needs to know whether it is running on a 6809 or a 6309 processor in a time sensitive application. No way of finding out in Basic09 but relatively easy for a ML subroutine which can pass its findings back to Basic09.

Hopefully you are by now convinced of the usefulness of this approach, so how to pull it off? As you may have noticed from my examples, all these applications are very straightforward and single purpose. This makes it much easier to write and debug the routine since it doesn't have to be very complicated.

For instance, if your routine has to deal with data from a variety of sources, your Basic09 program can standardize the data format and store the data in an internal buffer that can be accessed by the ML routine. Likewise Basic09 can take the results from your ML routine and process them some more if necessary.

Assuming you are somewhat familiar with writing assembly language, even under DECB assemblers like EDTASM, we are left with only one sticking point: how does Basic09 pass on variables?

Before I dig into that, there is one thing I want to clarify. I mentioned EDTASM be-

cause at this level (simple subroutines) you don't need to be an OS-9 expert to write ML routines. The only thing you have to keep in mind is that your code must be position independent in order to work correctly. For all practical purposes: use only BRANCH instructions and NO JUMP instructions and you won't have anything to worry about. Of course you will need a compiler or other program that can assemble a correct OS-9 format module around your code.

So how does Basic09 pass the necessary variables? Actually it is really simple. Basic09 creates a stack of 4-byte entries, sets the microprocessor's S register (hardware stackpointer) to point to the stack it just created and then transfers program control to the starting address of your ML routine.

That's really all there is to it. Returning control back to Basic09 is even simpler: just execute a RTS instruction. OS-9 finds the starting address by looking at two bytes inside the ML routine's module header but this, as well as some other overhead associated with starting up a module, is completely transparent to the user.

So all we are left with is to be able to interpret the values contained in the above mentioned stack. It's definitions are as follows: [please note that all values passed are integers (or 2-byte values) effectively passing 2 values per 4-byte entry] the first integer you encounter, the one at which S points, is an address Basic09 needs when your routine exits. So never change it or your program will crash the moment your ML routine exits. Also make sure that the S register points to this address when your routine exits for similar reasons. So be careful when using BSR or other instructions that change the stack pointer.

The second integer on the stack holds the number of variables that have been passed to the routine. For instance if you know that Basic09 must pass 3 variables, you can use the following code to check this out.

```
ldd 2,s    EC 62
cmpd #3    10 83 00 03
beq init   27 04
ldb #56    C6 38
coma      43
rts       39
init ...  start of rest of module
```

This code will return an error 56 to Basic09 if a different number of variables has been passed. Code 56 is defined as "parameter passing error" in Basic09.

The rest of the stack consists of a 4-byte entry for each variable passed. The lower

two bytes of the entry form a pointer to the variable, while the upper two bytes contain the size of the variable. You could use this data to make a further check, but other than that it is rarely used.

So our main focus becomes the actual pointers. The hard part about them is that Basic09 has more ways than one to pass a variable (pointer) to a subroutine and this is reflected in the meaning of the pointer on the stack. Basically it can be a direct pointer or an indirect pointer. As any ML programmer knows this means you have to use different addressing modes to get to the actual variable. The trick is to know when Basic09 passes a direct pointer and when an indirect one.

I will try to explain that without confusing everybody, including myself. Assuming that you are familiar with passing variables from one Basic09 module to another, one can safely say that if you use the same parameter passing conventions, most pointers to variables are indirect ones. Most pointers to arrays and datastructures, on the other hand, are direct pointers. Keeping this in mind let's see why.

```
DIM origin,destination,runlength:INTEGER
DIM buffer1(1000),buffer2(1000): BYTE
origin=ADDR(buffer1) \ destination=
ADDR(buffer2) runlength=500
10 RUN move(origin,destination,runlength)
20 RUN move(buffer1,buffer2, runlength)
```

Assuming the goal is to move 500 bytes from buffer1 to buffer2, let's see what happens. In line 10 Basic09 passes the addresses of both buffers as separate variables. In effect Basic09 passes a pointer to the address of the variable. To get the address of buffer1, which is the content of the variable "origin", we must treat the location pointed to by the value on the stack as an address. So we are dealing with an indirect pointer here, which requires indirect addressing.

Alternately, in line 20, Basic09 passes a pointer to buffer1 itself; rather than to a variable. This means that the memory location pointed to by the stack is the actual first byte of buffer1 and we must use direct addressing to access that byte.

So what instructions do we use to load the various registers with correct values? Let's say we want register x to point to buffer 1 and register y to buffer 2, so we can copy bytes using the auto-increment mode. The code used in line 20 is the more direct approach and easiest to understand.

```
Idx 4,s AE 64
ldy 8,s 10 AE 68
ldd [$c,s] EC F8 0C
```

The above code will load all your registers with the correct values.

In line 10 things are a little more complex

since we use indirect addressing to load registers x and y. This looks as follows.

```
Idx [4,s] AE F8 04
ldy [8,s] 10 AE F8 08
ldd [$c,s] EC F8 0C
```

This code, too, will load the correct values into your registers.

As you can see in both cases we loaded the number of bytes to be moved into register D. If it suits you better, you can also load this value into register U and use that as a loop counter.

Please note that every time you pass a variable like "runlength" to a ML routine you MUST pick it up using indirect addressing as explained above.

Finally, I would like to pass along some tips that will make things somewhat easier for those of you who want to try this approach: First of all if your ML routine runs successfully with some programs but refuses to run or repeatedly crashes your computer with new efforts: check the type of variable you are passing to the routine. Nine times out of ten you will find that they are not declared as INTEGER.

If you have to pass a considerable number of variables to a routine or if the routine needs a "scratchpad", try setting up a user stack with your Basic09 program: this is much easier than trying to do everything inside your ML routine. This user stack can be defined as a simple array of x bytes in size or as a complex datastructure, which can be preloaded with all initialization values by Basic09. Load the address of the array or data structure into register U and you're all set.

I wanted to finish this part of the series by including some code that can serve as an example. But what? After some consideration I decided to settle for a copying routine as this ties in nicely with some subjects I still want to discuss. Once you have such a routine you will find it becomes much easier to use data modules and/or graphics buffers to (temporarily) store data outside your 64K workspace. This way your programs can handle larger volumes without continually having to access your disk drives.

For those of you with an assembler I have included a listing with comments. For all others, here is the actual code including module headers and CRC bytes. You can save them as disk files in the same manner as described in part 3 of this series.

To avoid undue problems I have used the "indirect pointer" approach. This is a little more work, but more versatile and consistent throughout all situations. So, here is the Basic09 code you need to call the routine(s).

```
DIM org,dest,rle:INTEGER
org=ADDR(buffer1) \ dest=ADDR(buffer2)
rle=SIZE(buffer1)
```

RUN move(org,dest,rle)

[or RUN move(org,dest,rle)]

Of course buffer1 and buffer2 can be any variable or string name and rle holds whatever number of bytes you want to move. Keep in mind that any system calls used to map data blocks or buffers into your address space return a starting address for that block. You can access those blocks by equating org or dest to that starting address.

Now for the fun part: I will give you four different routines. One moves bytes, the second moves bytes in reverse order (which is sometimes necessary to avoid overwriting data before it is copied). The first set runs only on a 6809, the second set of routines do the same job on a 6309 using it's TFM instructions. After all, it would be cruel to tell you how fast the processor is without allowing access to it's capabilities. Although it should be obvious, the second two do not work with a 6809 processor (first two will work with 6309).

The last thing I want to mention here is that these routines do not check for errors. Your Basic09 program must make sure the addresses are valid and that RLE is greater than zero. If you feed the routine a negative runlength it will run but crash your program in the process because it overwrites vital sections of your address space.

```
move.6809
87 CD 00 36 00 2E 21 81 0D 00 0D 00
00 AE F8 04
10 AE F8 08 EC F8 0C ED 62 5D 27 0D
A6 80 A7 A0
5A 26 F9 A6 62 26 02 5F 39 4A A7 62
20 EE 6D 6F
76 E5 01 91 E4 22
```

```
rmove.6809
87 CD 00 3B 00 32 21 81 1C 00 0D 00
00 26 00 1D
30 01 10 AE F8 08 31 21 EC F8 0C ED
62 5D 27 0D
A6 82 A7 A2 5A 26 F9 A6 62 26 02 5F
39 4A A7 62
20 EE 72 6D 6F 76 E5 01 6F 04 0C
```

```
move.6309
87 CD 00 25 00 1D 21 81 2D 00 0D 00 00
00 AE F8 04
10 AE F8 06 10 A6 F8 0C 11 38 12 5F
39 6D 6F 76
E5 01 4E 95 F7
```

```
move.6309
87 CD 00 26 00 1D 21 81 2E 00 0D 00 00
AE F8 04
10 AE F8 06 10 A6 F8 0C 11 39 12 5F 39
72 6D 6F
76 E5 01 2A A0 4F
```

```

00001 NAM move.6809
00002 0001 edition equ 1
00003 0000 87CD0042000D mod
MODEND,MODNAM,MDTYPE+
LANG,$81,START,MO
00004 0020 MDTYPE equ $20
00005 0001 LANG equ $1
00006 0000 MODMEM equ 0
00007 000D 6D6F76E5 MODNAM fcs
/fmove/ 00008 * for rmove: MODNAM
fcs /fmove/
00008 * for rmove: MODNAM fcs /
rmove/
00009 0011 01 fcb edition
* 3 parameters?
00010 0012 EC62 START ldd 2,s
00011 0014 10830003 cmpd #3
00012 0018 2704 beq INIT
* no!! error 56
00013 001A C638 ldb #$38
00014 001C 43 coma
00015 001D 39 rts
00016 001E AEF804 INIT idx [4,s]
00017 * for rmove: leax 1,x avoid
loosing byte
00018 * in pre-decrement mode
00019 0021 10AEF808 ldy [8,s]
00020 * for rmove: leay 1,y same reason
00021 0025 ECF80C ldd [$c,s]
00022 0028 ED62 std 2,s scratchpad
00023 002A 5D tsbt rle mod 256=0??
00024 002B 270D beq next
00025 002D A680 loop lda ,x+
00026 * for rmove: loop lda ,-x
00027 002F A7A0 sta ,y+ copy bytes
00028 * for rmove: sta ,-y
00029 0031 5A decb countdown
00030 0032 26F9 bne loop
00031 0034 A662 lda 2,s more blocks
00032 0036 2602 bne next left to do?
* exit without errors
00033 0038 5F clr
00034 0039 39 rts
* decrement and store blockcount
00035 003A 4A next deca
00036 003B A762 sta 2,s
00037 003D 20EE bra loop
00038 003F 6E50AF EMOD
00039 0042 MODEND equ *

00000 error(s)
00000 warning(s)
$0042 00066 program bytes generated
$0000 00000 data bytes allocated
$0119 00281 bytes used for symbols

00001 NAM move.6309
00002 0001 edition equ 1
00003 0000 87CD0042000D mod
MODEND,MODNAM,MDTYPE+
LANG,$81,START,MO
00004 0020 MDTYPE equ $20

```

```

00005 0001 LANG equ $1 00006
0000 MODMEM equ 0
00007 000D 6D6F76E5 MODNAM fcs
/fmove/ 00008 * for rmove: MODNAM
fcs /fmove/
00009 0011 01 fcb edition
* 3 parameters?
00010 0012 EC62 START ldd 2,s
00011 0014 10830003 cmpd #3
00012 0018 2704 beq INIT
* no!! error 56
00013 001A C638 ldb #$38
00014 001C 43 coma
00015 001D 39 rts
00016 001E AEF804 INIT idx [4,s]
00017 0021 10AEF808 ldy [8,s]
00018 0025 10A6F80C ldw [$c,s]
00019 0029 113812 tfm x+,y+
00020 * for rmove: tfm x,-y-
* exit without errors
00021 002C 5F clr
00022 002D 39 rts
00023 002E 04F5B7 EMOD
00024 0031 MODEND equ *

00000 error(s)
00000 warning(s)
$0031 00049 program bytes generated
$0000 00000 data bytes allocated
$00D9 00217 bytes used for symbols

```

Chris Dekker
 RR #4
 Centreville, NB E0J 1H0
 CANADA

< 268'm>

Wildcard.bas listing

continued from page 19

```

1500 INPUT " TYPE";WN$ 
1510 FOR SI=0 TO NF
1520 :TS$=T$(SI)
1530 :IF WN$=TS$ THEN OC(SI)=ASC(OP$)
1540 NEXT
1550 RETURN
1560 'CASE A OR B
1570 INPUT " TYPE #(Y/N)";WT$ 
1580 IF WT$="Y" OR WT$="a" THEN
INP UT " TYPE";WT$:GOTO 1600
1590 IF WT$="N" OR WT$="n" THEN
WT $="N" ELSE GOTO 1570
1600 FOR SI=0 TO NF
1610 :TS$=T$(SI):AF$=A$(SI)
1620 :IF WT$<>"N" THEN IF WT$=TS$ AND WN$=AF$ THEN OC(SI)=ASC(OP$)
1630 :IF WT$="N" THEN IF WN$=AF$ TH ENOC(SI)=ASC(OP$)
1640 NEXT

```

```

1650 PRINT "DONE"
1660 RETURN
1670 'HELP
1680 CLS:PRINT:PRINT"help on wild cards"
1690 PRINT:PRINT " THE * MEANS USE
WILD CARDS "
1700 PRINT " FOR EXAMPLE TO MARK
ALL FILES"
1710 PRINT " THAT START WITH A C TY
PE C"
1720 PRINT " TO MARK ALL FILES THAT
START"
1730 PRINT " WITH CE TYPE CE* AND T
O MARK"
1740 PRINT " ALL FILES THAT CONTAIN
CE"
1750 PRINT " TYPE *CE"
1760 PRINT:PRINT
1770 INPUT " PRESS ANY KEY TO CONTI
NUE ",AK$
1780 RETURN
1790 'CONTAINS
1800 WL=LEN(WN$):WL=WL-1:WS$=RIG
HT$(WN$,WL):SF=0
1810 FOR SI=0 TO NF
1820 :FS$=F$(SI)
1830 :SF=INSTR(1,FS$,WS$)
1840 :IF SF<>0 THEN OC(SI)=ASC(OP$):
PRINT F$(SI);# ";SI
1850 NEXT
1860 PRINT "CONTAIN "+WS$+" PRESS
ANY KEY"
1870 K$=INKEY$:IF K$="" THEN GOTO 1870
1880 RETURN
1890 'STARTS WITH
1900 WL=LEN(WN$):WL=WL1:WS$=
LEFT $(WN$,WL)
1910 FOR SI=0 TO NF
1920 :FS$=LEFT$(F$(SI),WL)
1930 :IF FS$=WS$ THEN OC(SI)=ASC(OP
$): PRINT F$(SI);# ";SI
1940 NEXT
1950 PRINT "PRESS ANY KEY"
1960 K$=INKEY$:IF K$="" THEN
GOTO 1960
1970 RETURN
1980 'Mark ALL
1990 FOR I=0 TO NF
2000 :OC(I)=ASC(OP$)
2010 NEXT
2020 RETURN
2030 'RUN THE FILE ORGANIZER
2040 FOR I=0 TO NF
2050 :POKE 30200+I,OC(I)
2060 NEXT
2070 PRINT:PRINT "Done"
2080 PRINT "INSERT PROGRAM DISK IN
DRIVE";PD
2090 PRINT:PRINT " PRESS ANY KEY"
2100 WN$=INKEY$:IF WN$="" THEN 2100
2110 IF PD=0 THEN RUN"SDSK512K:0",R
2120 IF PD=1 THEN RUN"SDSK512K:1",R
2130 IF PD=2 THEN RUN"SDSK512K:2",R
2140 IF PD=3 THEN RUN"SDSK512K:3",R
2150 'HEADING
2160 CLS:PRINT " WILD CARD BUILDER
:PRINT " OP = ",OP$:PRINT
2170 RETURN

```

< 268'm>

micro notes

RiBBS to Go is a package of 5 dsdd 40 track disks with a reworked OS-9 boot disk from what you supply. To run it requires a CoCo III with 512k memory, OS-9 Level II, upgraded Multi-pak, RS232 pak or equivalent, and a minimum of 2 dsdd 40 track floppy drives. A minimum 1200 baud autoanswer modem is needed, but Hi-Speed modems can be accomodated for 9600 baud use. Connects at up to 16,800 will work as my ZyXEL modem will connect at 16,800 to another ZyXEL.

RiBBS to Go is the latest V2.10 and can easily be expanded to more menus, larger msg base and moved to a Hard Drive. Readme.info to do this is included along with 2 disks with all docs, and a set of utilities and games. The other 3 disks are the boot disk and a /dd CMDS and /d1/FILES disk (the running part). RiBBS to Go will run 'out of the box' with 15 min. of editing a few files; read-me files describe this procedure. It is also recommended to install an RS232 breakout box between your RS232pak and modem and rewire the pins 6-8 reversed between the pak and modem. RiBBS will work without this but not really the best way.

*** How to get RiBBS to Go ***

Format 5 DSDD 40 track 5-1/4" disks with OS-9 Lev II. Copy or cobbler one disk with a copy of your boot. Put in a floppy disk mailer or padded mail paket with your return address and a check or M.O. for \$5.00 for

News Flash!! CoCo 3 Emulator had been officially released!! As of today, November 26, 1994, Jeff Vavasour has officially released his CoCo 3 Emulator. As one of the Beta Testers for this emulator (and the only Beta Tester that has tested it for OS-9 Level II Compatibility), I have given my approval of the stability and usability of the emulator. The other testers have done the same for RS-DOS functionality.

One thing people should know about the emulator is that they should get the new CoCo 2 emulator (ver. 1.4) to run the VGACHECK program and test out the disk drive I/O before committing to registering the CoCo 3 emulator (VGA CHECK is on this issue's microdisk, OS-9 & DECB sides; it will require transferring to an MS-DOS disk before running).

CoCo 3 Emulator Features:

- Full CoCo 3 Basic and OS/9 Level II Compatibility
- Sound Card and PC Speaker support for CoCo DAC sound generation
- Use of mouse, or PC Joystick for either or both joystick ports
- Can use 40 or 80 track virtual disks in OS/9 Level II
- FULL REAL TIME MULTITASKING!! That's right - real multitasking on the PC!
- Printer fully functional in both DECB and

- King's Quest III
- Koronis Rift
- Rogue
- OS-9 Level II
- Multi-Vue
- DeskMate 3
- Sub-Etha's Towel & Invaders
- Most RS-DOS programs and utilities, such as Disk EDTASM, ColorVentures Backup Lightening, etc.

The CoCo 3 emulator is available only by registration. Registration info is found in the *new* ver 1.40 release of the CoCo 2 emulator, which should appear on SimTel FTP mirror sites shortly. The file is coco2-14.zip (*not* coco2-13.zip). One SimTel site is oak.oakland.edu, the directory is pub/msdos/emulator. When you register your copy of the CoCo 2 emulator, you will be sent the CoCo3 emulator disk. Once you have checked your MS-DOS system and verified that it meets the emulator requirements, you can send \$25 to:

Jeff Vavasour
c/o Department of Physics
6224 Agricultural Road
University of British Columbia
Vancouver, B.C.
Canada V6T 1Z1

Jeff's e-mail : jeffv@physics. ubc.ca

"micro notes" is for news on anything that may be of interest to readers, personal classified ads, and new product releases. If you see anything that may be of interest to other readers, have something for sale, or need an item, please write in and let us know! Vendors: be sure and let us know when you have a new product even if you don't already advertise!!

shipping and handling. The Package will be returned within a few days. Send to:

Warren Hrach
4369 Newport Ave.
San Diego, CA 92107

BBS (619) 224-4878 (RIBBS site!)
Fido net/node 1:202/744
Voice (619) 221-8246
Internet (warren@ocnbeach.jd.com)
A version of RiBBS is being worked for the MM/1 and K-Windows.

68' micros now has its own Internet news list! To subscribe to the list, send e-mail to "listserv@zog.com". The message text should read "subscribe micros firstname lastname", using your real first and last name. If you need a list of the commands supported by the list server, send another message to the same address with the message being the word "help". You will be sent a copy of the help file, usually by the next day. E-mail may also be sent to the editor at "microed@zog.com" as well as "dsrtfox@ Delphi.com".

OS-9 Level II

- All CoCo 3 Colors and video modes fully functional
- ROM, Virtual Disk, and Virtual Tape menus
- Improved Menu driven File Port Utility can be called up and used within emulator.

There is much more. This emulator will function just like the CoCo 3 and will do so right on your own PC compatible!

What you will need:

- PC Compatible (286 w/640K minimum required; 386SX/16 or higher recommended; at least 384K of EMS memory required for 512K emulation - 128K without EMS; VGA graphics.

- PC Sound Card (Optional - required for sound)
- Mouse (Joystick optional)
- CoCo 3 ROM and Disk Basic ROM from Disk Drive ROMPak (Instructions on getting the ROMs transferred from your disk based CoCo 3 system is included in the emulator package)

Some of the many Programs found to run with the emulator:

FOR SALE: Coco 3, excellent condition, with like-new CM-8 monitor, two DS floppies, all OS-9 public domain library, OS-9 levels 1 and 2, many, many programs and ROM packs, 2400bps external modem w/ 3 way serial port switch and cable, all manuals, Rainbow magazine from 1984 to final issue; Multi-Vue, 512K upgrade, 6309 w/socket and B&B PowerBoost software (not yet installed), joystick, X-pad, no HD or Multipak. E-mail at dclockman@pacifier.com, voice mail at 206-750-5266, voice at 206-896-4000, or drop a line to 15112 NE 34th Street, Vancouver, WA 98682-8315. \$300 obo takes all!

< 268'm >

*from my family to
yours, I wish you all a
very merry holiday
season and a happy
(and safe!) new year!*

the editor

NEW PRODUCTS from FARNA Systems!

CoCo Family Recorder/OS-9 1.0

If you are into genealogy, then the CoCo Family Recorder is the *absolute best* program for the CoCo 3. The OS-9 version is nearly identical to the DECB version in appearance, but takes advantage of many OS-9 features such as pop-up windows for entering data. DECB users can send their *original* CCFR disk (it will be returned) to get the OS-9 version for only \$20.00. Others must pay the regular price of \$28.50. Requires at least one 40 track double sided drive (FD-502) or larger. Can be shipped on 3.5" 720K disk if requested.

FARNA Systems
Box 321

Warner Robins, GA 31099

\$2.50 shipping and handling per order.
Canada S&H \$4.00; Overseas \$7.00

**NOTICE: WE CAN NOW
ACCEPT CREDIT CARDS!!!**
Visa and MasterCard (only!)
can now be accepted for payment.
There is a 6% service charge (but a
maximum of \$10) since cards are
accepted through a third party. Cards
will be billed through "FS Printing".

GREAT FIND!

We have recently purchased 10 **BRAND NEW** double sided forty track drives from a local vendor. He couldn't use them, so I can pass them to you at a used drive price of only \$15 each, shipping included! Remember, I only have ten, so act fast!



Box 321
Warner Robins, GA 31099
Phone 912-328-7859
Internet: dsrtfox@delphi.com

Software, Books, and Hardware for all OS-9/OSK Systems!

CoCo DECB Software:

CoCo Family Recorder - \$17.50
Genealogy program for CoCo 3. Requires 2 drives, 80 col. monitor.
NEW! OS-9 Version - \$32.50

DigiTech Pro - \$12.50

Sound recorder for CoCo3. Record any sound for easy play-back in your BASIC or M/L programs.

ADOS: Support for double sided drives, 40/80 tracks, faster formatting, much more!

Original (CoCo 1/2) - \$15.00
ADOS 3 (CoCo 3) - \$25.00
Extended ADOS 3 - \$30.00 (ADOS 3 req., RAM drives, support for 512K-2MB)
ADOS 3/Ext. Combo - \$50.00

Mind Games - \$7.50

Collection of 9 classic games. Run from included RAM disk w/512K.

Cross Road II - \$7.50

Simple Tic-Tac-Toe, but with amazing sound and graphics! Sound recorded with Digi-Tech.

Space Intruders - \$14.50

Looks just like Atari's classic "Space Invaders"! CoCo 1/2 and 3.

Donut Dilemma - \$14.50

Climb, jump, and ride elevators to top of Donut factory to shut it down! 10 levels. CC 1,2,3.

Rupert Rythym - \$14.50

Collect Rupert's stolen notes, then work out correct sequence. Great action adventure! Get Space Intruders, Donut Dilemma, and Rupert Rythym for only \$33.50! Save \$10!

CoCo OS-9 Software:

Patch OS-9 - \$7.50
Automated program installs most popular/needed patches for OS-9 Level II. 512K and two 40T/DS (or larger) drives required. (128K /35T users can install manually- state 35T.)

OS-9 Point of Sale - \$62.50

Maintain inventory, print invoices, customer catalog, etc. Multi-user capable under Level II. Supports ASCII terminals. Basic09 required. Simple menu driven interface.

Books:

Tandy's Little Wonder - \$22.50
140 page softbound book with history and technical info for all CoCo models. Schematics, peripherals, upgrades, modifications, repairs, much more- all described in detail! Vendors, clubs, BBSs also listed.

OS-9 Quick Reference Guides Level II (Revision 2) - \$7.50

68K (based on 2.3) - \$10.50
Get that bulky manual off your desk! These handy QRGs have all the most needed information in a 5.5" x 8.5" desk-top size. Includes command syntax, error codes, special keys functions, etc.

CoCo Hardware:

DigiScan Video Digitizer - \$150
Capture images from VCR, camcorder, or TV camera. No MPI required- uses joystick ports. CoCo Max3, Max 10, Color Max 3 compatible. Special order- allow 90 days for delivery. Send \$75 deposit.

Ken-Ton SCSI Hard Drive System and Components

Controller only	\$135.00
OS-9 Drivers	\$25.00
RGB-DOS (for DECB access)	\$35.00
** \$50 for RGB-DOS and OS-9 drivers when purchased together with a controller **	
"Y" cable, \$25 for two position, \$35 for three.	
Drive cables - specify direct to drive or SCSI case type connector	\$25.00

Add \$2.50 S&H per order. Canada/Mexico add \$4.00; Overseas \$7.00

FARNA Systems Publishing Services

Type Setting and Printing: We can prepare professional typeset manuals, books, booklets, catalogs, and sales flyers for you - we can print or you reproduce as needed from a master set! Very reasonable prices - inquire!

Mailing Service: If you send catalogs or letter correspondence to 200 or more persons at once, we can do all work for you for about the same cost of your materials alone! How much is your time worth???

Contact Frank Swygert at above address/phone for quotes

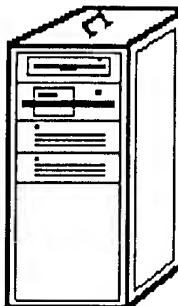
for all your CoCo hardware needs, connect with

CoNect

449 South 90th Street
Milwaukee, WI 53214
E-mail: rickuland@delphi.com

NEW PRODUCT: The Cube

This tower enclosure was designed specifically for the CoCo and peripherals, even an MPI if desired! Four drive bays, two will hold a pair of 3.5" drives sideways. Easy access, carry handle mounted on top!



Mini RS-232 Port: Don't let the name fool you! This is a full featured serial port, supporting the signals needed for flow control as well as the basic 4. Jumper blocks allow readdressing or swapping DSR/DCD. No custom cables or hardware widgets needed here! Y cable users will need to add \$9.95 for a power supply. **\$49.95**

Hitachification: CoNect will install a Hitachi 63B09E CPU and a socket into your CoCo. Machine MUST be in working condition! The 68B09E will be returned unharmed. 90 day limited warranty. Chip and installation only **\$29.95**

REPAIRS: We can repair most damaged CoCos, even those with bad traces where a 68B09 was removed. Costs vary with damage. Bad 68B09 sockets repaired for only **\$40!** Inquire BEFORE sending your computer.

By arrangement with StrongWare, Sub-Etha Software is proud to bring you...

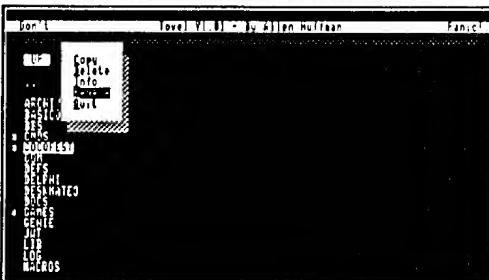
Soviet Bloc - Called the best Tetris(tm)-like game by many, this is a bigger version of the now-classic falling shapes puzzle game. RS-DOS Req: CoCo 3, Joystick/Orchestra-90 Pak optional **\$19.95**

GEMS - "Columns" of colors fall as you change the order of the colors. Match three in a row, column, or diagonal at the bottom and those colors disappear. Sounds simple, doesn't it? RS-DOS Req: CoCo 3, Joystick/Orchestra-90 Pak optional **\$24.95**

Copy Cat - Simon says "match the sequence of tones as the colored diamonds flash". Great for building memory skills. RS-DOS Req: CoCo 3 **\$ 9.95**
OSK Req: MM/1 or 100% K-Windows Systems **\$14.95**

HFE (Hprint Font Editor) - A fantastic editor for those HPRINT fonts with lots of options. Create your own character set which you can LOAD and use in your own programs. Also creates and edits fonts compatible with MiniBanners! RS-DOS Req: CoCo 3, Joystick optional **\$19.95**

Font Collection - A collection of 18 useable HPRINT or MiniBanners fonts. (Importable to OS-9 for use with MiniBanners9 as well!) RS-DOS Req: **\$ 9.95**



Towel! V1.01 by Allen Huffman - NEW VERSION! A program no intergalactic hitchhiker should be without! Use mouse or keyboard hot-keys to perform common disk and file commands from pull-down menus. Tag multiple files for Copy, Delete, Rename, etc., and even point 'n click a Backup, Cobbler, Dcheck, or whatever. User definable menu for custom options. Runs under the EthaWin interface (included) on a high-speed text screen. All commands/colors configurable. OS9 Req: CoCo 3, OS-9 Level 2 **\$19.95**

Sub-Etha Software P.O. Box 152442 Lufkin, TX 75915 Add \$2.50 S&H. Texas residents add 8.25% tax. Write us for more info!

DISTO Products

Quality hardware for your Color Computer!

2MB Upgrade (no RAM) - **\$99.95**

Mini Disk Controller - **\$70**

Super Controller I - **\$100**

Super Controller II - **\$130**

MPROM Burner - **\$50**

3-N-1 (parallel, RS-232, RTC) - **\$75**

SASI/SCSI Hard Disk Adapter - **\$75**

Full Turn of the Screw - book with all

Tony DiStefano's Rainbow articles,

January 1983 to July 1989 - **\$20**

Complete Schematic Set - all DISTO product schematics except 2MB upgrade - **\$20**

Include \$2 S&H for book or schematics.

Hardware S&H is \$4.50 for one item, \$6.50 two or more. Certified check or International Money Order only!

Running low on some items- please call for availability!

514-747-4851

DISTO
1710 DePatie,
St. Laurent, QC H4L 4A8
CANADA

Sub-Etha Originals:

MultiBoot V1.03 by Terry Todd & Allen Huffman - Type "DOS" and bring up a list of up to sixteen OS9BOOT files! No more floppy swapping. A serious must-have for intense OS-9 users. OS9 Req: CoCo 3, OS-9 Level 2 **\$19.95**

1992 CoCoFest SIMULATOR V1.02 by Allen Huffman - NEW VERSION now uses compressed graphics and has scoring. Take a walking tour of the '92 Atlanta CoCoFest with this graphics adventure. 16-level digitized photos of the event and a text parser (ie, "get the box of disks") to let you interact. Runs on a 640x192 graphics screen. OS9 Req: 512K CoCo 3, OS-9 Level 2, 500K+ of Disk Space **\$ 9.95**
OSK Req: MM/1 or 100% K-Windows System **\$14.95**

Write-Right by Joel Mathew Hegberg - Featureful word processor for the MM/1 with what you would expect from a "real" word processor. What you see is what you get! **\$54.95**

Etha-GUI by Joel Mathew Hegberg - A neat program launcher for the MM/1 which includes handy desktop utilities like a phone dialer and nifty screen savers...even a trash can. Point and click icons to run programs. **\$34.95**

Quality OS-9 Software from

ColorSystems

NEW! K-Windows Chess for MM/1

Play chess on your MM/1.....\$24.95

NEW! X-10 Master Control for MM/1

Use MM/1 to control you home!.....\$29.95

Variations of Solitaire

Pyramid, Klondike, Spider, Poker and Canfield
MM/1.....\$29.95 CoCo3.....\$19.95

OS-9 Game Pack

Othello, Yahtzee, KnightsBridge, Minefield,
and Battleship
MM/1.....\$29.95 CoCo3.....\$19.95

WPShell

An OS-9 Word Processing Point and Click Interface
CoCo3.....\$14.95

Using AWK with OS-9

Includes V2.1.14 of GNU AWK for OS-9/68000
MM/1.....\$14.95

To order send check or money order to:

Color Systems
P.O. Box 540
Castle Hayne, NC 28429
(916) 675-1706

Call or write for a free catalog! Demo disks also available.

NC Residents please add 6% sales tax

Owned and operated by Zack C. Sessions

Summertime is "off" season for a lot of CoCoists. If you are one of those, look forward to new releases and upgrades this fall. If you use your CoCo all year 'round, the following titles are currently available:

CoCoTop version 1.0 \$24.95
CoCoTop version 1.1 \$19.95
CoCoTop 1.1 + Tools 3 \$34.95
OScopy/RScopy \$10.00
TOOLS 3 version 1.1 \$29.95
Quickletter version 2.0 \$19.95
Accounting level 2 \$34.95
Investing level 2 \$24.95
Level II graphics 1.2 \$34.95

upgrades only \$5.00 (return original disk)

Shipping+handling: US/Canada \$3.00 all others \$5. Prices in US dollars. Send cheque or money order NO COD'S. Call or write for Canadian dollar prices. Mention the name of this magazine in your order and you will receive a free bonus disk!

C. Dekker ... User-friendly Level II Programs!

RR #4 Centreville, NB

EOJ 1H0, CANADA

Phone 506-276-4841



EDTASM6309 Version 2.02 ----- \$35.00

This is a major patch to Tandy's Disk EDTASM to support Hitachi 6309 codes. Supports all CoCo models. CoCo 3 version uses 80 column screen, 2MHz. YOU MUST ALREADY OWN TANDY'S DISK EDTASM TO MAKE USE OF THIS PRODUCT. It WILL NOT work with a disk patched cartridge EDTASM.

CC3FAX ----- \$35.00

Extensive modification to WEFAX (Rainbow, 1985) for 512K CoCo 3. Uses hi-res graphics, holds full 15 min. weather fax image in memory. Large selection of printer drivers. Requires shortwave receiver and cassette cable (described in documentation)

HRSDOS ----- \$25.00

Move programs and data between DECB and OS-9 disks. Supports RGB-DOS for split DECB/OS-9 hard drives. No modifications to system modules (CC3Disk or HDisk) required.

DECB SmartWatch Drivers ----- \$20.00

Access your SmartWatch from DECB! New function added to access date/time from BASIC (DATE\$). Only \$15.00 with any other purchase!

RGBBOOST ----- \$15.00

Make the most of your HD6309 under DECB! Uses new 6309 functions for a small gain in speed. Compatible with all programs tested to date! Only \$10.00 with any other purchase!

Robert Gault

832 N. Renaud

Grosse Pointe Woods, MI 48236

313-881-0335

Add \$4 shipping & handling per order

Peripheral Technology

Specials

486SLC/33MHz Motherboard w/CPU \$139.00

486SLC/50MHz IBM, ISA,CPU, 0K \$199.00

486SLC/66MHz IBM, VESA, CPU, Math \$299.00

IBM boards - Made in USA - 3YR warranty

1MB SIMM 70ns DRAM \$47.00

356MB Samsung IDE Drive \$229.00

420MB Connor IDE Drive \$275.00

546MB Maxtor IDE Drive \$379.00

IDE/Floppy/Serial/Parallel \$24.95

1.44MB TEAC Floppy \$49.95

Mini Tower, 200W, LED readout \$79.00

Panasonic Dual Speed CD ROM \$169.00

VGA Card ET4000-1MB, 1280x1024 \$99.00

VGA Monitor WEN 28mm 1024x768 \$249.00

UPS Ground \$7.00 on most items except Tower & monitor (\$12.00 UPS Ground).

1250 E. Piedmont Rd. 404/973-2156

Marietta, GA 30062 FAX: 404/973-2170

The OS-9 User's Group, Inc.

Working to support OS-9 Users

Membership includes the Users Group newsletter, MOTD, with regular columns from the President, News and Rumors, and "Straight from the Horse's Mouth", about the use of OS-9 in Industrial, Scientific and Educational institutions.

Annual Membership Dues:
 United States and Canada Other Countries
 25.00 US 30.00 US

The OS-9 Users Group, Inc.
 6158 W. 63d St. Suite 109
 Chicago, IL 60638
 USA

*From our
family to
yours, please
have a very
merry and safe
Christmas and
New Year!*

Northern Xposure *'Quality Products from
North of the Border'*

OS-9 Level II Color Computer 3 Software	<hr/>
NitrOS-9 v1.20 Call or write for upgrade info or new purchase procedure. Requires Hitachi 6309 CPU	\$29.95
Shanghai:OS-9 Introductory price	\$25.00
Send manual or RomPak to prove ownership	
Thexder:OS-9 Send manual or RomPak to prove ownership	\$29.95
Smash! Breakout-style arcade game	\$29.95
Rusty Launch DECB/ECB programs from OS-9!	\$20.00
Matt Thompsons SCSI System v2.2 'It flies!'	\$25.00
256/512 byte sectors, multipak support	

Disk Basic Software	<hr/>
Color Schematic Designer v3.0 New lower price	\$30.00
Oblique Triad Software	<i>Write for catalogue</i>

Color Computer 3 Hardware	<hr/>
Hitachi 6309 CPU (normally 'C' model, may be 'B')	\$15.00
SIMM Memory Upgrade Runs Cooler! 512k \$44.95 0K \$39.95	
Sound Digitizing cable	\$15.00

OS-9/68000 Software	<hr/>
OSTerm 68K v2.2 External transfer protocol support	\$50.00
TTY/ANSI/VT100/K-Windows/Binary Emulation	
Upgrade from TasCOM (Send TasCOM manual please)	\$30.00

7 Greenboro Cres *All prices in U.S. funds.*
 Ottawa, ON K1T 1W6 *Check or MO only.*
 CANADA *Prices include S&H*
 (613)736-0329
 Internet mail: cmckay@northx.isis.org

ADVERTISER'S INDEX:

<i>BlackHawk Enterprises</i>	18
<i>C. Dekker</i>	28
<i>Chicago CoCo Fest</i>	6
<i>Color Systems</i>	28
<i>CoNect</i>	27
<i>Delmar Company</i>	BC
<i>DISTO</i>	27
<i>FARNA Systems</i>	21, 26
<i>J&M Microtek</i>	21
<i>Northern Xposure</i>	29
<i>OS-9 User's Group</i>	29
<i>Robert Gault</i>	28
<i>Small Grafx Etc.</i>	6
<i>Sub-Etha Software</i>	29

Don't have a subscription yet?

WHAT ARE YOU WAITING FOR?!

Subscribe today!

(Details inside front cover)

For superior OS-9 performance, the

SYSTEM V

Provides a 68020 running at 25 MHz, up to 128 MBytes of 0 wait-static memory, SCSI and IDE interfaces, 4 serial and 2 parallel ports, 5 16-bit and 2 8-bit ISA slots and much more. The SYSTEM V builds on the design concepts proven in the SYSTEM IV, providing maximum flexibility with inexpensive expandability. Now available at 33 MHz.

AN OS-9 FIRST - the MICROPROCESSOR is mounted on a daughter board which plugs onto the motherboard. This will permit inexpensive upgrades in the future when even greater performance is required.

G-WINDOWS benchmark performance index is 0.15 seconds faster with a standard VGA board than a 68030 running at 30 MHz with ATTC video board (85.90 seconds vs 86.05 seconds).

Or, for less demanding requirements, the

SYSTEM IV

The perfect low cost, high-quality and high performance OS-9 computer serving customers worldwide. Designed for and accepted by industry. Ideal low-cost work-station, development platform, or just plain fun machine. Powerful, flexible, and inexpensively expandable. Uses a 68000 microprocessor running at 16 MHz.

Both computers provide flexible screen displays in the native mode with the optional VGA card.

Eight text modes are supported: 40 x 24, 80 x 25, 80 x 50, 100 x 40, 132 x 25, 132 x 28, 132 x 44, and 132 x 60. Foreground, background, and border colors are user selectable from up to 16 colors.

Eleven graphics modes are supported: 640 x 200 x 16, 320 x 200 x 256, 640 x 350 x 16, 640 x 350 x 256, 40 x 480 x 16, 640 x 400 x 256, 800 x 600 x 16, 640 x 480 x 256, 1024 x 768 x 16, 800 x 600 x 256, and 1024 x 768 x 256.

Text and graphics modes may be selected by utility provided, MODESET, by software using SetStcals, or by termcap entries. In the text mode, the screen responds to standard VT100 control sequences. The full character set from Hex 20 through Hex FF is supported in the text modes up to and including 100 characters wide. The upper 128 characters follow the IBM Character Set 2 popular with many terminals and printers. These may be displayed on the screen by using the 'Alt' key and one or two other keys (software permitting).

Optional G-WINDOWS provides three screen resolutions: 640 x 480 x 256, 800 x 600 x 256, or 1024 x 768 x 256. You can have two full size 80 x 25 windows with room to spare. Or, a window as large as 122 x 44 using the large fonts or one over 180 x 70 using the small fonts.

delmar co

PO Box 78 - 5238 Summit Bridge Road - Middletown, DE 19709
302-378-2555 FAX 302-378-2556